Springer Tracts in Advanced Robotics

Volume 67

Editors: Bruno Siciliano · Oussama Khatib · Frans Groen

Daniel Schütz and Friedrich M. Wahl (Eds.)

Robotic Systems for Handling and Assembly

Springer

Professor Bruno Siciliano, Dipartimento di Informatica e Sistemistica, Università di Napoli Federico II, Via Claudio 21, 80125 Napoli, Italy, E-mail: siciliano@unina.it

Professor Oussama Khatib, Artificial Intelligence Laboratory, Department of Computer Science, Stanford University, Stanford, CA 94305-9010, USA, E-mail: khatib@cs.stanford.edu

Professor Frans Groen, Department of Computer Science, Universiteit van Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands, E-mail: groen@science.uva.nl

Editors

Dipl.-Ing. Daniel Schütz
Technische Universität
Braunschweig
Institute of Machine Tools
and Production Technology
Langer Kamp 19b
38106 Braunschweig
Germany
E-mail: d.schuetz@tu-bs.de

Prof. Friedrich M. Wahl
Technische Universität
Braunschweig
Institute for Robotics
and Process Control
Mühlenpfordtstraße 23
38106 Braunschweig
Germany
E-mail: f.wahl@tu-bs.de

ISBN 978-3-642-16784-3 e-ISBN 978-3-642-16785-0

DOI 10.1007/978-3-642-16785-0

Springer Tracts in Advanced Robotics ISSN 1610-7438

Library of Congress Control Number: 2010937997

© 2010 Springer-Verlag Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable for prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

5 4 3 2 1 0

springer.com

Editorial Advisory Board

Oliver Brock, TU Berlin, Germany
Herman Bruyninckx, KU Leuven, Belgium
Raja Chatila, LAAS, France
Henrik Christensen, Georgia Tech, USA
Peter Corke, Queensland Univ. Technology, Australia
Paolo Dario, Scuola S. Anna Pisa, Italy
Rüdiger Dillmann, Univ. Karlsruhe, Germany
Ken Goldberg, UC Berkeley, USA
John Hollerbach, Univ. Utah, USA
Makoto Kaneko, Osaka Univ., Japan
Lydia Kavraki, Rice Univ., USA
Vijay Kumar, Univ. Pennsylvania, USA
Sukhan Lee, Sungkyunkwan Univ., Korea
Frank Park, Seoul National Univ., Korea
Tim Salcudean, Univ. British Columbia, Canada
Roland Siegwart, ETH Zurich, Switzerland
Gaurav Sukhatme, Univ. Southern California, USA
Sebastian Thrun, Stanford Univ., USA
Yangsheng Xu, Chinese Univ. Hong Kong, PRC
Shin'ichi Yuta, Tsukuba Univ., Japan

STAR (Springer Tracts in Advanced Robotics) has been promoted under the auspices of EURON (European Robotics Research Network)

Foreword

Robotics is undergoing a major transformation in scope and dimension. From a largely dominant industrial focus, robotics is rapidly expanding into human environments and vigorously engaged in its new challenges. Interacting with, assisting, serving, and exploring with humans, the emerging robots will increasingly touch people and their lives.

Beyond its impact on physical robots, the body of knowledge robotics has produced is revealing a much wider range of applications reaching across diverse research areas and scientific disciplines, such as: biomechanics, haptics, neurosciences, virtual simulation, animation, surgery, and sensor networks among others. In return, the challenges of the new emerging areas are proving an abundant source of stimulation and insights for the field of robotics. It is indeed at the intersection of disciplines that the most striking advances happen.

The Springer Tracts in Advanced Robotics (STAR) is devoted to bringing to the research community the latest advances in the robotics field on the basis of their significance and quality. Through a wide and timely dissemination of critical research developments in robotics, our objective with this series is to promote more exchanges and collaborations among the researchers in the community and contribute to further advancements in this rapidly growing field.

The volume edited by Daniel Schütz and Friedrich M. Wahl provides a unique collection of a sizable segment of the robotics research in Germany. It reports on contributions from eight institutes in Braunschweig brought together within ten years by the Collaborative Research Center on Robotic Systems for Handling and Assembly (SFB 562) funded by the German Research Foundation.

This twenty-five-chapter volume covers important research topics on parallel kinematic robots, an area which has received a great deal of attention by the research community in view of the advantages of robots based on closed kinematic chains over the traditional serial industrial robots. The contents are organized in four parts, covering theoretical and computational aspects

of modeling, system design, control and programming as well as the existing and upcoming technology and components for implementation of such robots.

The thorough discussion, rigorous treatment, and wide span of the work unfolding in this area reveal the significant advances in the theoretical foundation and technology basis of parallel kinematic robots and their promising potential in various application domains. SFB 562 culminates with this important reference to the world robotics community on the current developments and new directions undertaken by this team of German researchers. A fine addition to the STAR series!

Naples, Italy
August 2010

Bruno Siciliano
STAR Editor

Preface

This volume presents research results of the Braunschweig research endeavor in the field of parallel robotics. In the summer of 2000, the German Research Foundation (DFG) established the Collaborative Research Center 562 (Sonderforschungsbereich 562 or in short: SFB 562) to conduct basic and application oriented research to develop fundamental concepts and solutions to overcome several drawbacks of the more traditional serial industrial robots. As is well known, robots based on closed kinematic chains constitute a promising alternative to conventional serial robot structures, especially with respect to dynamics, precision, and stiffness. Despite the continuously increasing interest of the robotics community in parallel kinematic robots (PKRs), the commercial market is still in a development stage. Ten years of extensive research within the SFB 562 has improved the understanding of the underlying problems and certainly created substantial solutions to important open issues.

Eight institutes of the Electrical Engineering, Mechanical Engineering, and Computer Science Faculties of the Technische Universität Braunschweig and the Institute of Composite Structures and Adaptive Systems of the German Aerospace Center in Braunschweig participated in the SFB 562 project. Most of the research results have been published in numerous national and international journals as well as at renowned conferences. Some major results have been presented at the three International Colloquia on Robotic Systems for Handling and Assembly held together with leading international experts in Braunschweig in 2003, 2005 and 2008. This book summarizes the major findings of the project. It is organized in four chapters as described in the following paragraphs:

Part I: Kinematics, Dynamics – Modeling and Design

The Denavit-Hartenberg representation is a well-known and widely used convention for describing serial kinematic structures. To make this representation also applicable to parallel and hybrid structures, U. Thomas et al.

extend it in their paper 'A Unified Notation for Serial, Parallel and Hybrid Kinematic Structures'. The paper 'Structure and Type Synthesis of Parallel Manipulators' by M. Frindt et al. lays the foundation for structure synthesis as well as optimization of PKRs with respect to given application requirements. M. Rose shows in his paper how efficient real-time code for inverse dynamic models of parallel robots can be automatically generated. There are numerous publications on different parallel kinematic structures in the open literature. C. Stechert et al. propose how to exploit this knowledge by means of a knowledge-based methodology and tools for the constructional design of PKRs. F. Dietrich et al. present a geometry- and physics-based method for detecting and avoiding singularities of PKRs. In order to drive PKRs to their precision limits, sophisticated calibration principles are required. P. Last et al. propose new concepts of self-calibration and compare them to already known approaches.

Part II: Implemented Systems

One of the major aims of the SFB 562 was a thorough evaluation of the theoretical findings - not only by simulations but also through extensive experiments. To achieve this objective, several prototypes of PKRs have been designed and implemented. Our goal was to reach a superior level of professionalism for our machines. The paper by D. Schütz et al. gives an overview of the realized PKRs and discusses their impressive performance features. As a basic theoretical and experimental platform, K. Stachera et al. show in their paper 'Modelling, Control and Evaluation of an Experimental Adaptronic Five-Bar Robot' how adaptronic machine elements can be advantageously integrated in PKRs, e.g., in order to suppress vibrations arising resulting from high accelerations or decelerations. The paper by S. Kock et al. shows how redundancy can be exploited to substantially improve the performance figures of PKRs. S. Algermissen et al. show in their contribution how to decompose the workspace of the robot in order to realize a robust and stable gain scheduling for smart (i.e. adaptronic) structures by switching among a set of optimized controllers. The workspace limitations of PKRs have commonly been considered as a disadvantage. The contribution of Ch. Budde et al. shows how several robot configurations corresponding to different working and assembly modes can be used to substantially enlarge the workspace.

Part III: Control and Programming

PKRs, especially if adaptronic components are integrated, pose high demands on control approaches and the underlying control architecture. In order to meet the hard real-time requirements of PKRs and at the same time provide a clear, modular and easy-to-comprehend programming interface to robot control system developers, a generic homogeneous middleware for high-speed distributed real-time communication has been proposed by Finkemeyer et al. In order to increase the real-time performance for PKRs, this middleware

has been augmented with shared memory mechanisms as well as a dedicated industrial automation protocol, as described in the contribution by Y. Dadji et al. Integrated force and motion control in unconstrained and constrained space is extensively studied in the papers by M. Kolbus et al. and T. Reisinger et al. R. Osypiuk et al. introduce stiffness actuators based on parallel kinematic structures which are either attached to robot hands or installed in the environments. By applying active and highly dynamic impedance control, the maximum forces and torques between the manipulators and their environments caused by impact can be significantly decreased. T. Kröger et al. introduce a new programming paradigm as interface between sensor-based motion control and robot programming, so-called manipulation primitives, which thoroughly exploit Masons pioneering work on compliant motion control in the early 1980s. Based on this paradigm, F. Dietrich et al. describe the control architecture of the SFB 562 PKRs including design patterns for active connectors, modular motion planning, and sensor integration. The dream of automated robot application programming has not yet become true, but important steps towards it in the domain of robotized assembly have been developed by U. Thomas et al. in their paper on automated robot programming. The last two papers in this section by J. Steiner et al. deal with self-management mechanisms and model based quality assurance aspects in complex robotic software architectures.

Part IV: Adaptronics and Components

As mentioned previously, the SFB 562 project aimed at a high level of professionalism when realizing the experimental prototype PKRs. When we started the project, we found that critical machine elements were not available on the market, and even worse, had not yet been addressed by the research community. This especially applies to the adaptronic components we developed for PKRs in order to suppress vibrations or to gain precision. The paper by M. Rose 'Modelling of Piezoceramic Patches for Augmenting Modal Structural Models with Flat Actuator Devices' shows how such devices in combination with suitable control algorithms can increase structural damping significantly. The contribution of R. Keimer et al. discusses design and implementation issues of adaptronic robot components. One further critical machine element for building PKRs are joints. N. Pavlovic et al. focus on optimized conventional passive joints and adaptive joints with integrated piezo actuators; the latter can be adapted to different operation conditions. Last but not least, M. Kirchhoff et al. propose two new microtechnologically fabricated inductive angle sensors and show how to integrate them in PKR joints. Joint integrated microsensors proved to be a useful means for self-calibration, speeding up the DKP and workspace monitoring of PKRs.

This summer, the project will come to an official end after 10 years of research. Nevertheless, there are still open issues which deserve further investigation in the near future. Many results contained in this monograph

are certainly of high value to engineers and scientists dealing with parallel robotics. But we are also convinced that there are many important and useful results which are relevant for the robotics community in a broader sense.

Braunschweig, Germany,
June 2010

Daniel Schütz
Project Manager
Friedrich M. Wahl
Chairman of the SFB 562

Acknowledgements

The editors of this monograph would like to express their deep thankfulness to all contributing persons for their outstanding high engagement in a very productive and warm atmosphere. This applies to the participating researchers, the diploma, bachelor, and master students involved in our projects as well as to our technicians and our secretaries, who together supplied the SFB 562 with an excellent technical and administrative infrastructure. Special thanks also go to our colleagues Profs. Albert Albers, Michael Braun, Jürgen Gausemeier, Bodo Heimann, Günter Hommel, Roland Kasper, Paul Levi, Lothar Litz, Günther Schmidt, Gerhard Schweitzer, Günther Seliger and Albert Weckemann who repeatedly acted as project reviewers. Numerous colleagues from outside Braunschweig stimulated our research during many fruitful discussions on various occasions. The editors are also indebted to the editors of the Springer Tracts in Advanced Robotics and their reviewers. Moreover, we are grateful to Dr. Ditzinger and his team from Springer-Verlag for managing and realizing the print of this monograph in an accustomed high quality.

Last but not least we would like to express our sincere gratitude to Drs. Andreas Engelke, Susanne Grindel, Anne Lipp, Wolfgang Rohe, Ruth Schellberg and Klaus Wefelmeier for accompanying the project in a constructive and cooperative manner. Without the financial support of the DFG this project would not have been possible!

Contents

Part I: Kinematics, Dynamics – Modeling and Design

A Unified Notation for Serial, Parallel and Hybrid Kinematic Structures 3
Ulrike Thomas, Friedrich M. Wahl

Structure and Type Synthesis of Parallel Manipulators 17
Matthias Frindt, Mathias Krefft, Jürgen Hesselbach

Automated Generation of Efficient Real-Time Code for Inverse Dynamic Parallel Robot Models 39
Michael Rose

Knowledge-Based Design Principles and Tools for Parallel Robots 59
Carsten Stechert, Hans-Joachim Franke, Thomas Vietor

Detection and Avoidance of Singularities in Parallel Kinematic Machines 77
Franz Dietrich, Jochen Maaß, Carlos Bier, Ingo Pietsch, Annika Raatz, Jürgen Hesselbach

Calibration of Parallel Kinematic Structures – Overview, Classification and Comparison 93
Philipp Last, Annika Raatz, Jürgen Hesselbach

Part II: Implemented Systems

Parallel Kinematic Structures of the SFB 562 109
Daniel Schütz, Christoph Budde, Annika Raatz, Jürgen Hesselbach

Modeling, Control, and Evaluation of an Experimental Adaptronic Five-Bar Robot 125
Krzysztof Stachera, Frank Schreiber, Walter Schumacher

Redundant Parallel Kinematic Structures and Their Control 143
Sönke Kock, Walter Schumacher

Robust Gain Scheduling for Smart-Structures in Parallel Robots 159
Stephan Algermissen, Michael Sinapius

Configuration Switching for Workspace Enlargement 175
Christoph Budde, Manfred Helm, Philipp Last, Annika Raatz, Jürgen Hesselbach

Part III: Control and Programming

A Middleware for High-Speed Distributed Real-Time Robotic Applications 193
Bernd Finkemeyer, Torsten Kröger, Friedrich M. Wahl

A Communication Architecture for Distributed Real-Time Robot Control 213
Yannick Dadji, Harald Michalik, Nnamdi Kohn, Jens Steiner, Guido Beckmann, Tobias Möglich, Jörn-Uwe Varchmin

Integrated Force and Motion Control of Parallel Robots – Part 1: Unconstrained Space 233
Michael Kolbus, Frank Wobbe, Thomas Reisinger, Walter Schumacher

Integrated Force and Motion Control of Parallel Robots – Part 2: Constrained Space 253
Thomas Reisinger, Frank Wobbe, Michael Kolbus, Walter Schumacher

Parallel Stiffness Actuators with Six Degrees of Freedom for Efficient Force/Torque Control Applications 275
Rafal Osypiuk, Torsten Kröger

Manipulation Primitives—A Universal Interface between Sensor-Based Motion Control and Robot Programming 293
Torsten Kröger, Bernd Finkemeyer, Friedrich M. Wahl

RCA562: Control Architecture for Parallel Kinematic Robots 315
Franz Dietrich, Jochen Maaß, Annika Raatz, Jürgen Hesselbach

Assembly Planning and Task Planning—Two Prerequisites for Automated Robot Programming 333
Ulrike Thomas, Friedrich M. Wahl

Self-management within a Software Architecture for Parallel Kinematic Machines 355
Jens Steiner, Ursula Goltz, Jochen Maaß

Model Based Quality Assurance for a Robotic Software Architecture 373
Jens Steiner, Karsten Diethers, Matthias Hagner, Ursula Goltz

Part IV: Adaptronics and Components

Modelling of Piezoceramic Patches for Augmenting Modal Structural Models with Flat Actuator Devices 393
Michael Rose

Design and Implementation of Adaptronic Robot Components 413
Ralf Keimer, Michael Sinapius

Passive and Adaptive Joints for Parallel Robots 429
Nenad Pavlović, Robert Otremba, David Inkermann, Hans-Joachim Franke, Thomas Vietor

Design and Implementation of New Sensors and Their Integration in Joints 445
Maren Ramona Kirchhoff, Jens Güttler, Alexander Wogersien, Nenad Pavlović, Robert Otremba, Hans-Joachim Franke, Stephanus Büttgenbach

Author Index 461

List of Contributors

Stephan Algermissen
German Aerospace Center (DLR)
Institute of Composite
Structures and
Adaptive Systems
Lilienthalplatz 7
38108 Braunschweig
Germany
stephan.algermissen@dlr.de

Guido Beckmann
Beckhoff Automation GmbH
Eiserstraße 5
33415 Verl
Germany
g.beckmann@beckhoff.de

Carlos Bier
IAV GmbH
Rockwellstraße 16
38518 Gifhorn
Germany
carlos.cezar.bier@iav.de

Christoph Budde
ABB Corporate Research Center
Wallstadter Straße 59
68526 Ladenburg
Germany
christoph.budde@de.abb.com

Stephanus Büttgenbach
Technische Universität Braunschweig
Institute for Microtechnology
Alte Salzdahlumer Straße 203
38124 Braunschweig
Germany
s.buettgenbach@tu-bs.de

Yannick Dadji
Technische Universität Braunschweig
Institute of Computer and
Communication Network
Engineering
Hans-Sommer-Straße 66
38106 Braunschweig
Germany
dadji@ida.ing.tu-bs.de

Karsten Diethers
Capgemini sd & m AG
Carl-Wery-Straße 42
81739 München, Germany
karsten.diethers@
capgemini-sdm.com

Franz Dietrich
Technische Universität Braunschweig
Institute of Production Automation
and Machine Tools
Langer Kamp 19b
38106 Braunschweig, Germany
f.dietrich@tu-bs.de

Bernd Finkemeyer
KUKA Roboter GmbH
Zugspitzstraße 140
86165 Augsburg
Germany
berndfinkemeyer@kuka-roboter.de

Hans-Joachim Franke
Technische Universität Braunschweig
Institute for Engineering Design
Langer Kamp 8
38106 Braunschweig, Germany
franke@ikt.tu-bs.de

Matthias Frindt
Technische Universität Braunschweig
Institute of Production Automation
and Machine Tools
Langer Kamp 19b
38106 Braunschweig
Germany
m.frindt@tu-bs.de

Ursula Goltz
Technische Universität Braunschweig
Institute for Programming and
Reactive Systems
Mühlenpfordtstraße 23
38106 Braunschweig
Germany
goltz@ips.cs.tu-bs.de

Jens Güttler
Technische Universität Braunschweig
Institute for Microtechnology
Alte Salzdahlumer Straße 203
38124 Braunschweig
Germany
j.guettler@tu-bs.de

Matthias Hagner
Technische Universität Braunschweig
Institute for Programming and
Reactive Systems
Mühlenpfordtstraße 23
38106 Braunschweig
Germany
hagner@ips.cs.tu-bs.de

Manfred Helm
Volkswagen AG, Dept. 1564
Berliner Ring 2
38442 Wolfsburg, Germany
manfred.helm@volkswagen.de

Jürgen Hesselbach
Technische Universität Braunschweig
Institute of Production Automation
and Machine Tools
Langer Kamp 19b
38106 Braunschweig, Germany
j.hesselbach@tu-bs.de

David Inkermann
Technische Universität Braunschweig
Institute for Engineering Design
Langer Kamp 8
38106 Braunschweig, Germany
inkermann@ikt.tu-bs.de

Ralf Keimer
German Aerospace Center (DLR)
Institute of Composite
Structures and
Adaptive Systems
Lilienthalplatz 7
38108 Braunschweig, Germany
ralf.keimer@dlr.de

Maren Ramona Kirchhoff
Technische Universität Braunschweig
Institute for Microtechnology
Alte Salzdahlumer Straße 203
38124 Braunschweig, Germany
m.kirchhoff@tu-bs.de

Sönke Kock
ABB Corporate Research
Forskargränd 7
72178 Västerås, Sweden
soenke.kock@se.abb.com

Nnamdi Kohn
IAV GmbH
Rockwellstraße 16
38518 Gifhorn, Germany
nnamdi.kohn@iav.de

Michael Kolbus
Technische Universität Braunschweig
Institute of Control Engineering
Hans-Sommer-Straße 66
38106 Braunschweig
Germany
kolbus@ifr.ing.tu-bs.de

Mathias Krefft
Technische Universität Braunschweig
Institute of Production Automation
and Machine Tools
Langer Kamp 19b
38106 Braunschweig
Germany
m.krefft@tu-bs.de

Torsten Kröger
Stanford University
Artificial Intelligence Laboratory
Department of Computer Science
Stanford, CA 94305-9010, USA
t.kroeger@tu-bs.de

Philipp Last
Siemens AG
Dept. E D MV C R & D 4
Nonnendammallee 104
13629 Berlin
Germany
philipp.last@siemens.com

Jochen Maaß
SkySails GmbH & Co.KG
Veritaskai 3
21079 Hamburg
Germany
j.maass@tu-bs.de

Harald Michalik
Technische Universität Braunschweig
Institute of Computer and
Communication Network
Engineering
Hans-Sommer-Straße 66
38106 Braunschweig, Germany
michalik@ida.ing.tu-bs.de

Tobias Möglich
Technische Universität Braunschweig
Institute of Computer and
Communication Network
Engineering
Hans-Sommer-Straße 66
38106 Braunschweig
Germany
moeglich@ida.ing.tu-bs.de

Rafal Osypiuk
Technische Universität
Braunschweig
Institute for Robotics and
Process Control
Mühlenpfordtstraße 23
38106 Braunschweig
Germany
rafal.osypiuk@zut.edu.pl

Robert Otremba
Technische Universität
Braunschweig
Institute for Engineering Design
Langer Kamp 8
38106 Braunschweig
Germany
otremba@ikt.tu-bs.de

Nenad Pavlović
Institute for Engineering Design
Technische Universität Braunschweig
Langer Kamp 8
38106 Braunschweig
Germany
pavlovic@ikt.tu-bs.de

Ingo Pietsch
BSH GmbH
Corporate Technology
Innovation and Technology
Management (ZTI)
Carl-Wery-Straße 34
81739 München
ingo.pietsch@bshg.com

Annika Raatz
Technische Universität Braunschweig
Institute of Production Automation
and Machine Tools
Langer Kamp 19b
38106 Braunschweig
Germany
a.raatz@tu-bs.de

Thomas Reisinger
ABB Corporate Research Center
Wallstadter Straße 59
68526 Ladenburg
Germany
thomas.reisinger@de.abb.com

Michael Rose
German Aerospace Center (DLR)
Institute of Composite
Structures and
Adaptive Systems
Lilienthalplatz 7
38108 Braunschweig
Germany
michael.rose@dlr.de

Frank Schreiber
Technische Universität
Braunschweig
Institute of Control Engineering
Hans-Sommer-Straße 66
38106 Braunschweig
Germany
schreiber@ifr.ing.tu-bs.de

Walter Schumacher
Technische Universität
Braunschweig
Institute of Control Engineering
Hans-Sommer-Straße 66
38106 Braunschweig
Germany
w.schumacher@ifr.ing.tu-bs.de

Daniel Schütz
Technische Universität Braunschweig
Institute of Production Automation
and Machine Tools
Langer Kamp 19b
38106 Braunschweig
Germany
d.schuetz@tu-bs.de

Michael Sinapius
German Aerospace Center (DLR)
Institute of Composite
Structures and
Adaptive Systems
Lilienthalplatz 7
38108 Braunschweig
Germany
michael.sinapius@dlr.de

Krzysztof Stachera
Technische Universität Braunschweig
Institute of Control Engineering
Hans-Sommer-Straße 66
38106 Braunschweig
Germany
stachera@ifr.ing.tu-bs.de

Carsten Stechert
Technische Universität Braunschweig
Institute for Engineering Design
Langer Kamp 8
38106 Braunschweig
Germany
stechert@ikt.tu-bs.de

Jens Steiner
Technische Universität Braunschweig
Institute for Programming and
Reactive Systems
Mühlenpfordtstraße 23
38106 Braunschweig
Germany
steiner@ips.cs.tu-bs.de

Ulrike Thomas
German Aerospace Center (DLR)
Institute of Robotics and
Mechatronics
P.O. Box 1116
82234 Weßling
Germany
ulrike.thomas@dlr.de

Jörn-Uwe Varchmin
Technische Universität Braunschweig
Institute of Electrical Measurement
and Fundamental Electrical
Engineering
Hans-Sommer-Straße 66
38106 Braunschweig
Germany
j-u.varchmin@tu-bs.de

Thomas Vietor
Technische Universität Braunschweig
Institute for Engineering Design
Langer Kamp 8
38106 Braunschweig
Germany
vietor@ikt.tu-bs.de

Friedrich M. Wahl
Technische Universität
Braunschweig
Institute for Robotics and
Process Control
Mühlenpfordtstraße 23
38106 Braunschweig
Germany
f.wahl@tu-bs.de

Frank Wobbe
Technische Universität
Braunschweig
Institute of Control Engineering
Hans-Sommer-Straße 66
38106 Braunschweig
Germany
wobbe@ifr.ing.tu-bs.de

Alexander Wogersien
Technische Universität Braunschweig
Institute for Microtechnology
Alte Salzdahlumer Straße 203
38124 Braunschweig
Germany
a.wogersien@tu-bs.de

Part I
Kinematics, Dynamics – Modeling and Design

A Unified Notation for Serial, Parallel and Hybrid Kinematic Structures

Ulrike Thomas and Friedrich M. Wahl

Abstract. In this contribution we present a uniform notation for any kind of kinematic structure ranging from serial robots and parallel robots to hybrid kinematic structures as well as from multi-finger grippers to the kinematics of locomotive or humanoid robots. If kinematic structures contain passive joints, they often are of spherical or cardan nature. To describe these types of joints, a new notation based on the well-known Denavit-Hartenberg notation is presented. Additionally, closed kinematic chains and structures with more than one chain attached to a robot basis can be denoted by our graph based representation. Our goal is to provide the robot community with a unified description of kinematic structures - as it has been done by the classical DH-parameter notation for serial robots - in order to support the development for exchangeable programming tools and ideas.

1 Introduction

Alongside with flight simulators, which are mainly based in the Gough-Stewart platform [1, 2], many other parallel mechanisms have been invented throughout the past decades. As it is widely known, parallel robots own a different kinematic structure compared to their serial counterparts. They consist of closed kinematic chains, which lead to some favorable properties such as high stiffness, high accuracy and low moved masses. Due to this, high accelerations and high velocities can be achieved and at the same time high precisions can be reached with parallel

Ulrike Thomas
German Aerospace Center (DLR), Institut of Robotics and Mechatronics,
P.O.Box 1116, 82234 Weßling, Germany
e-mail: ulrike.thomas@dlr.de

Friedrich M. Wahl
Technische Universität Braunschweig, Institute for Robotics and Process Control,
Mühlenpfordtstraße 23, 38106 Braunschweig, Germany
e-mail: f.wahl@tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 3–15.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

robots. Thus, in handling and assembly an increasing amount of applications can be found, in which parallel robots are employed [3]. One of the first high speed parallel kinematic machines was the *Delta*-structure, which is used in production technology up to now [4, 5]. Additional to the development of new parallel kinematic machines, much research work concentrates on the analysis of the workspace including singularity analysis, workspace optimization and optimal structural design e.g. [6, 7, 8, 9]. Moreover a lot of work deals with solving the direct kinematic problem of such structures, which still for many structures is an open problem. For some planar 3-DOF structures and for Stewart platforms analytical solutions are known, but still for others there are only numerical solutions available [10, 11]. In almost every application descriptions regarding the kinematic structure as well as dynamics parameters are indispensable. Still, many scientists use their own individual notation for assigning the corresponding parameters to parallel manipulators, depending on the problems they face, and on the structures they describe. In addition, many software tools for parallel robots concerning control, collision free path planning, and simulation are being developed. In [12] we have proposed a new unified kinematic description of parallel or hybrid kinematic structures containing spherical and cardan joints. In this paper, we remind on the importance of a uniform description of kinematic structures. For a notation of serial chain segments inside hybrid kinematic structures, Denavit-Hartenberg parameters are widely applied [13]. In parallel and hybrid manipulators spherical and cardan joints are frequently involved. Describing these joint types by DH-parameters leads to a non-unique representation. Furthermore, notating e.g. spherical joints by DH-parameters results in 3 necessary 4-parameter sets. When developing software tools for manipulators many engineers from various fields are involved, hence, denoting a particular joint and its associated joint variables should be defined well. At the beginning of the Collaborative Research Center 562 (SFB 562)[1] in the year 2000, no widely used common notation simultaneously suitable for serial, parallel, and hybrid manipulators has been established. Thus, there existed a strong demand for a uniform notation for any kind of kinematic structures, which might be as useful as the Denavit-Hartenberg notation for serial manipulators. This paper proposes a notation, which extends the well-known DH-parameters for denoting spherical and cardan joints in an analogous way. It also provides a solution for representing and referring to the kinematic structure of parallel and hybrid manipulators in a unique manner. When specifying such a notation, following requirements have to be taken into consideration:

- *COMPLETENESS:* The specification of the notation should provide all kinematic and geometric data of a given structure and vice versa, i.e., a bijective mapping between structures and notations exists.
- *UNIQUENESS:* An instantiated notation for a particular kinematic structure should refer only to this structure.
- *NON-REDUNDANCY:* Multiple notations of a particular property should be avoided. Each property has to be represented once in the specification.

[1] http://www.tu-braunschweig.de/sfb562

- *COMPREHENSIVENESS:* The notation should be expandable, for instance to describe dynamic parameters, as well.

In the following, we propose a notation fulfilling these requirements while simultaneously reducing the number of parameters, necessary to describe spherical or cardan joints. This notation has been used throughout the years of developing algorithms and tools for our parallel robots by the team of the Collaborative Research Center 562.

2 Related Work

Mayer and Gosselin [14] use a notation to describe a general Gough-Stewart platform in which they refer to joints by their layers. For describing spherical joints, they employ roll, pitch and yaw (RPY) parameters. Merlet and Gosselin [11] classify parallel kinematic structures into SSM (Simplified Symmetric Manipulators), TSSM (Triangular Simplified Symmetric Manipulators) and MSSM (Minimal Simplified Symmetric Manipulators). In [15] a closed forward position solution is given for a PPSP manipulator. The authors describe spherical joints by three sets of DH-parameters. Merlet [7] employs a notation for spherical joints by three coplanar rotational joints. The notation proposed by us is based on the approach by Belifiore and Benedetto [16], who apply graphs to represent serial, parallel, and hybrid redundant kinematic structures with one degree of freedom at each joint. Graphs as shown there, have also been applied in the field of mechanisms and gear trains [17, 18]. For denoting spherical joints we augment the DH-parameters as described in this paper. Veitschegger et al. [19] mentioned also an extension of the DH-parameter defintion, but just for calibration purposes. In contrast to our proposal, Veitschegger uses a rotation around the y-axis, which is multiplied at the end of the four DH-transformations.

3 Description of Joints with More Than One Degree of Freedom

Joints with one degree of freedom can be described in a very intuitive manner by DH-parameters (Fig. 1) where, one transformation ${}^{i-1}\boldsymbol{T}_i$ is assigned to each robot link:

$$ {}^{i-1}\boldsymbol{T}_i = Rot(\boldsymbol{z}_{i-1}, \theta_i) \cdot Trans(\boldsymbol{z}_{i-1}, d_i) \cdot Trans(\boldsymbol{x}_i, a_i) \cdot Rot(\boldsymbol{x}_i, \alpha_i) \tag{1} $$

Imagine a cylindiracl joint – a combined revolute and prismatic joint – has two DOFs with one rotational and one translational DoF, it can also be denoted by this equation with θ_i and d_i being the joint variables. Other joints with more than one DOF can either be described by

- several sets of DH-parameters
- or by a more complex transformation.

The advantages of the second approach are at first a convenient kind of formulation and at second a reduction of redundancies. Furthermore for each joint exist as many

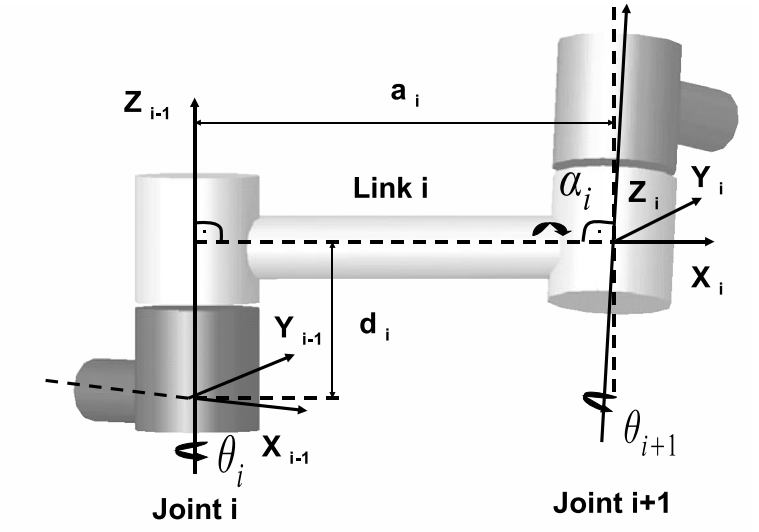


Fig. 1. The well known Denavit-Hartenberg Parameters.

joint variables as DOFs. Therefore, we propose to describe joints with more than one DOF by augmenting the DH-parameters.

3.1 Notation of Spherical Joints

In the literature, spherical joints are either described by RPY-parameters or by Euler angles. Using these descriptions, the representation is obviously ambiguous, because there exists more than one possible combination of three angles to reach a desired orientation of a spherical or even a cardan joint, see Fig. 2. Thus, some agreements for the usage of parameters are indispensable; moreover a top down compatibility to the well-known DH-parameters should be ensured. In our approach we extend the DH-parameters, so that it is possible to obtain three joint variables for a single spherical joint. For control purposes it is not essential to specify passive spherical joints by three joint variables, but e.g., for simulation we have to deal with all three DOFs explicitly. Consider the kinematic chain with three joints depicted in Fig.3.

Each spherical joint is characterized by its center point. Furthermore, one can connect these points to obtain normal vectors between the spherical joints. The cross product $\boldsymbol{n}_i \times \boldsymbol{n}_{i-1}$ of the succeeding and preceding joint normals is assigned to be the z-axis; the normal obtained by connecting the center points itself represents the x-axis, and the y-axis is obtained by applying the right hand rule. The x-axis points toward the succeeding joint. Using this convention, it is possible to assign coordinate systems to spherical joints in a non-ambiguous manner. We assume that the rules described here are not applied in singular poses of joints. Therefore, a

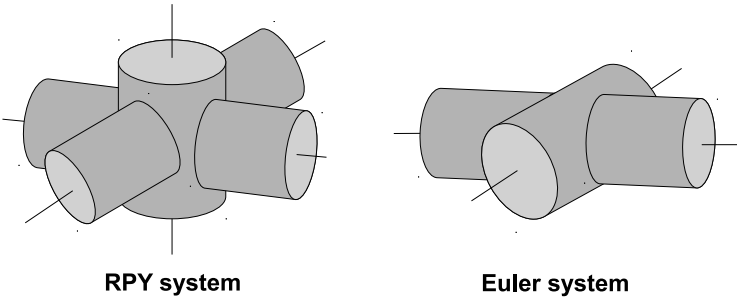


Fig. 2. Some common ambiguous representations.

single extended transformation ${}^{i-1}\boldsymbol{T}_i$ for a spherical joint consists of five elementary transformations defined by five parameters $\theta, d, \beta, a, \alpha$ respectively:

$$
\begin{aligned}
{}^{i-1}\boldsymbol{T}_i = Rot(\boldsymbol{z}_{i-1}, \theta_i) \cdot Trans(\boldsymbol{z}_{i-1}, d_i) \cdot Rot(\boldsymbol{y}'_{i-1}, \beta_i) \\
\cdot Trans(\boldsymbol{x}_i, a_i) \cdot Rot(\boldsymbol{x}_i, \alpha_i)
\end{aligned}
\tag{2}
$$

Thus, the DH-parameters have to be augmented by a rotation around the y'-axis, which is the transformed y-axis by the first two transformations. Fig. 4 shows an RSR (revolute, spherical, revolute)-chain, where each transformation step is depicted for the spherical joint. The first parameter θ_i describes a rotation around the z_{i-1}-axis, d_i in this case is zero, and the third parameter β_i is the rotation around the y'_{i-1}-axis, so that the new x-axis lies collinear to the normal vector of the link. The translation along the x_i-axis with parameter a_i is the distance between the

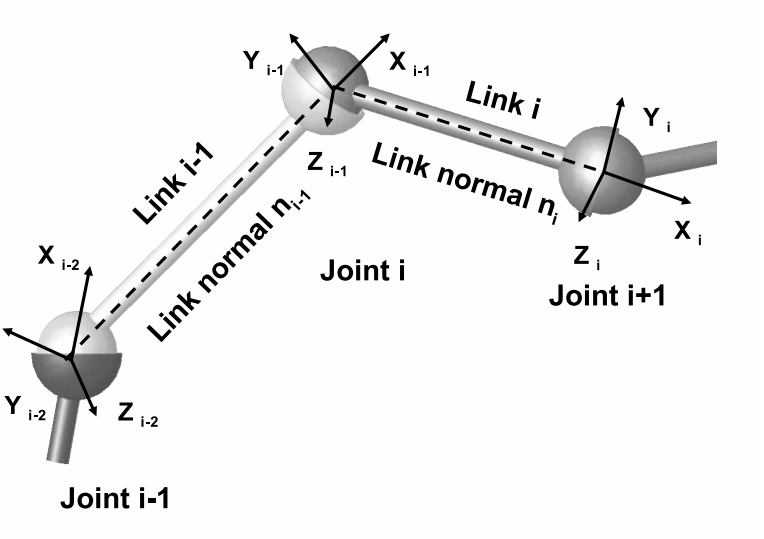


Fig. 3. The definition of axes for spherical joints.

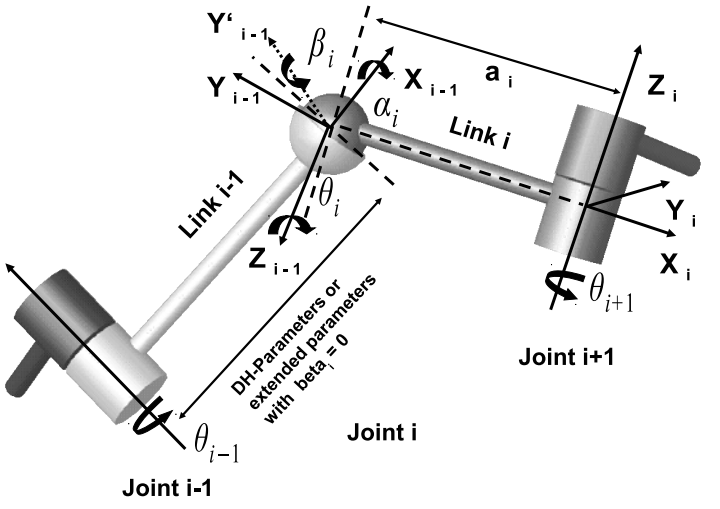


Fig. 4. The extended DH-parameters for a spherical joint inside an RSR-chain.

origins of the corresponding coordinate systems. The last rotation around the x-axis determines the third joint variable α_i. With this transformation the coordinate system which corresponds to the robot link i moves around a sphere with radius a_i. Using this definition, we are able to describe kinematic chains, which consist of a combination of rotational, prismatic or/and spherical joints with a maximum of five parameters for each robot link. For rotational and prismatic joints we set $\beta_i = 0$ and obtain the familiar DH-parameters. Hence, in order to describe revolute joints or prismatic joints θ_i and d_i are the joint variables respectively.

3.2 Cardan Joints

Cardan joints can also be described by the notation introduced above, see Fig. 5.The two DOFs are represented by joint variables θ_i and β_i. The x-axis of the corresponding coordinate system lies collinear to the normal of the link and points toward joints with higher indices.

3.3 Conventions for a Unique Extended DH-Definition

In order to define the coordinate systems uniquely some conventions are necessary:

- The origin of the coordinate system i corresponding to link i is located in the joint $i + 1$.
- The z-axis of the coordinate system i corresponds to the rotational or prismatic joint axis of joint $i + 1$. For spherical joints, the z-axis is assigned to the cross product of the joint normal vectors $\boldsymbol{n}_i$ and $\boldsymbol{n}_{i+1}$, assuming not $n_i \| n_{i+1}$.

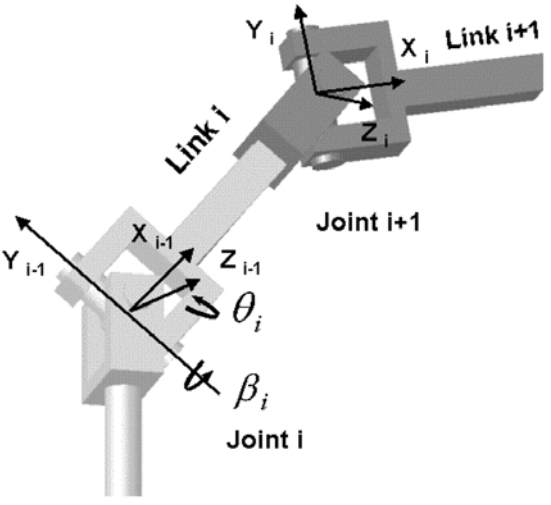
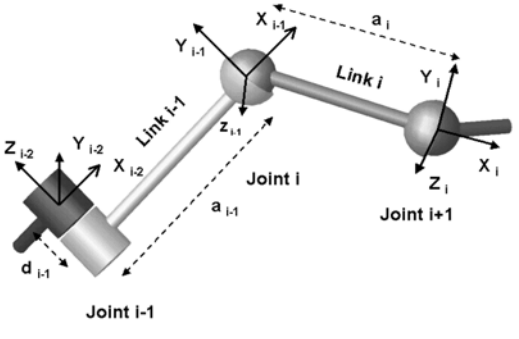


Fig. 5. Coordinate systems assigned for a hypothetical chain of cardan joints.

- The x-axis of link i lies collinear to the joint normal and points toward joints with higher indices.
- The rotational DOF must be instantiated from the left side in the transform equation above, i.e., if a joint has only DOF, θ_i is the joint variable, if it has two DOFs, θ_i and β_i are the variables. For spherical joints θ_i, β_i, and α_i with $0 \leq \beta_i \leq 180$ are the joint variables. The limitation is necessary in order to obtain an unambiguous specification.

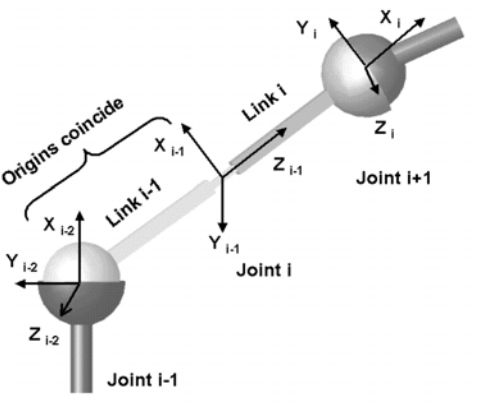


Joints	θ	d	β	a	α
$i-1$	*var*	d_{i-1}	0	a_{i-1}	90°
i	*var*	0	*var*	a_i	var
$i+1$	*var*	0	*var*	a_{i+1}	var

Fig. 6. An RSS (revolute, spherical, spherical)-chain and its coordinate systems according to the extended DH-parameter sets.

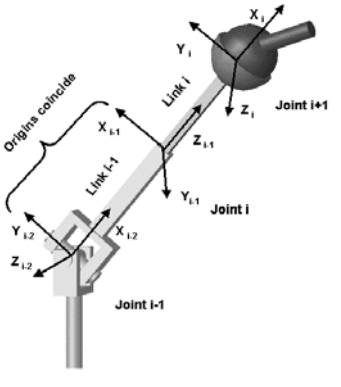
3.4 Examples of the Extended DH-Notation

In Fig. 6, 7 and 8 some examples of kinematic chains and their extended DH-parameter notations are shown, which are typically used in the design of parallel robots.



Joints	θ	d	β	a	α
$i-1$	*var*	0	*var*	0	*var*
i	−90°	*var*	90°	0	0
$i+1$	var	0	var	a_{i+1}	var

Fig. 7. An SPS (spherical, prismatic, spherical)-chain and its coordinate systems according to the extended DH-parameter sets.



Joints	θ	d	β	a	α
$i-1$	*var*	d_{i-1}	*var*	0	90°
i	−90°	*var*	−90°	0	0
$i+1$	var	0	var	a_{i+1}	*var*

Fig. 8. A CPS (cardan, prismatic, spherical)-chain and its coordinate systems respectively to the extended DH-parameter sets.

4 Notation of a Universal Robot

A notation of kinematic structures should be applicable to any kind of kinematic structure ranging from serial, parallel and hybrid robots to multi-finger grippers, locomotive robots and humanoid robots. Fig. 9 shows a *Hexa* platform and its kinematic structure. The directed graph on the right side can be obtained, if we map robot links to nodes of the graph and robot joints to graph edges. By this, links with multiple connections are represented by nodes in a non-ambiguous manner. If we map joints to nodes and links to edges, we would obtain an ambiguous definition of the tool and the base platform.

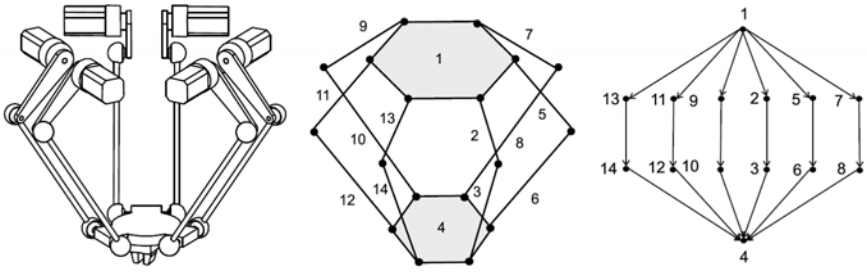


Fig. 9. The kinematic structure of the *Hexa*-robot and its respective graph representation on the right.

Formally, a kinematic graph or a kinematic net of a universal robot UR can be described by the following definition.

Definition 1. *A universal robot UR is defined as* 4*-tuple by* $UR := \langle \mathcal{N}, \mathcal{E}, \mathcal{BF}, \mathcal{TF} \rangle$*, where $\mathcal{N}$ is the set of nodes, $\mathcal{E}$ is the set of edges, $\mathcal{BF}$ and $\mathcal{TF}$ are used for specifying the robot tool and the base frame. The kinematic graph is attached by the following attributes:*

- $\mathcal{N} := \langle n_1, \ldots, n_n \rangle$ is a set of nodes. A node represents a physical connection between at least two joints. It corresponds to a robot link and consequently to its geometric entity. A node is defined by a tuple $n_i := \langle Slink | Mlink | Base | Tool, CSG_i \rangle$. $Slink$ means one connection between two joints, $Mlink$ represents multiple connections to more than two joints and $Base$ and $Tool$ are used for the base and tool platform respectively. CSG_i is the constructive solid geometry model or any other CAD-representation for the robot link i.
- $\mathcal{E} := \langle e_1, \ldots, e_m \rangle$ is a set of edges. Each edge is unidirectional and represents a joint, which connects two robot links. Such an edge exists for each joint. The edges $e_i := \left\langle \mathcal{N}_{pre} \times \mathcal{N}_{post}, \boldsymbol{T}_{fix,1}, \mathcal{T}_{params}, \boldsymbol{T}_{fix,2}, Type, Range, Flag \right\rangle$ are attributed by the following items:

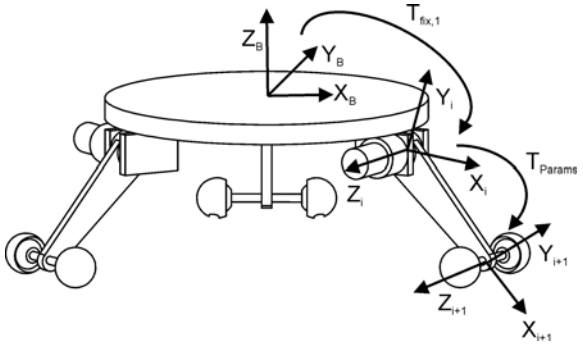


Fig. 10. The three homogeneous transformations for one chain.

- $\mathcal{N}_{pre} \times \mathcal{N}_{post}$ are pre and post nodes
- $\boldsymbol{T}_{fix,i} \in \Re^{4\times4}$ are fixed homogenous transformations. They are used, for example, to transform the robot base frame into the frame attached to the first link of a single chain, see Fig. 10.
- $\mathcal{T}_{params} := \langle \theta, d, \beta, a, \alpha \rangle$ describes the parameters for a transformation ${}^{i-1}\boldsymbol{T}_i$ according to the extended DH-parameter notation introduced above. The product $\boldsymbol{T}_{fix,1} \cdot {}^{i-1}\boldsymbol{T}_i \cdot \boldsymbol{T}_{fix,2}$ describes the transformation of a single edge. The matrix $\boldsymbol{T}_{fix,1}$ is equal to the identity matrix $\boldsymbol{I}$ for the transformation from the base frame to the first joint and respectively the matrix $\boldsymbol{T}_{fix,2}$ is set to $\boldsymbol{I}$ for the transformation from the last joint to the tool. In all other cases $\boldsymbol{T}_{fix,1}$ and $\boldsymbol{T}_{fix,2}$ are the identity matrices $\boldsymbol{I}$, see Fig. 10.
- $Type := \langle revolutional | prismatic | spherical | cardan | 2d - linear | \ldots \rangle$ describes the type of the joint.
- *Range* defines the range of each joint.
- $Flag := \langle passive, active \rangle$ is used to mark an active or passive joint.

- $\mathcal{BF} := \langle \boldsymbol{F}, n_i \rangle$ with $\boldsymbol{F} \in \Re^{4\times4}$ is the base frame attached to the base platform represented by the node n_i.
- $\mathcal{TF} := \left\langle \boldsymbol{F}, n_j \right\rangle$ with $\boldsymbol{F} \in \Re^{4\times4}$ is the tool frame attached to the tool platform represented by node n_j.

With the notation introduced here, we are able to describe serial, parallel, and hybrid kinematic structures in a systematic and intuitive way. In any case we obtain a directed graph. Fig. 11 shows the kinematic layout of the *Eclipse*-robot [15] and its corresponding directed graph. Fig. 12 demonstrates the use of our representation for the *Portys*-robot [3], and Fig. 13 depicts the application of the graph representation for the *Delta*-robot. By applying our new specification, we are able to deal with all kinds of kinematic structures using one common representation. Here, we have demonstrated the usability for several different robot structures.

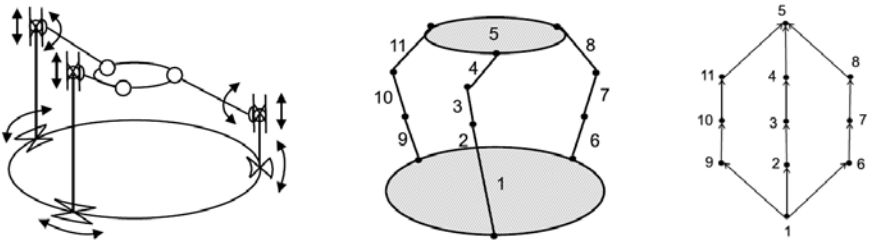


Fig. 11. The *Eclipse*-robot [15] its layout and its corresponding graph representation.

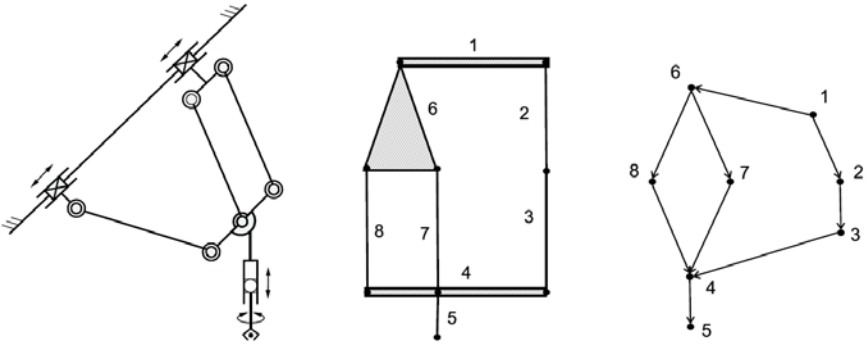


Fig. 12. The *Portys*-robot [3] its layout and its corresponding graph representation.

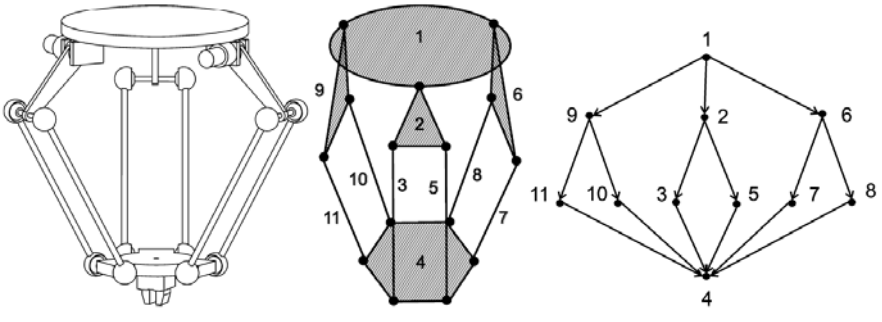


Fig. 13. The *Delta*-robot [4] its layout and its corresponding graph representation.

4.1 An Algorithm for the Enumeration of Nodes

In the case when different modules, e.g. a simulator, a motion planner, or a control module, have to exchange data, i.e. joint vectors, a systematic way is desired to refer to links and joints in the kinematic graph. The following algorithm assigns indices to nodes, such that each node obtains a unique number.

1. Unmark all nodes.
2. Start with the node to which the base frame is attached.
3. Assign the current index to the visited node.
4. Sort all unmarked neighbors such that the neighbor with the geometrically shortest distance $\boldsymbol{d}_i$ is the first item in the list etc. The distance between two adjacent link coordinate systems ξ and η in their home position is defined by $\boldsymbol{d}_i = \|(x_\xi - x_\eta, y_\xi - y_\eta, z_\xi - z_\eta)^T\|$ or in case of zero by $\boldsymbol{d}_i = \|(\varphi_{x\xi} - \varphi_{x\eta}, \varphi_{y\xi} - \varphi_{y\eta}, \varphi_{z\xi} - \varphi_{z\eta})^T\|$. ξ and η represent the transformation annotated by $\boldsymbol{T}_{fix,1} \cdot {}^{i-1}\boldsymbol{T}_i \cdot \boldsymbol{T}_{fix,2}$.
5. Increment index.
6. For all unmarked successors continue with step 3.
7. Stop if all nodes have been visited.

This algorithm searches the nodes in depth first order and assigns indices according to the criteria above.

5 Conclusion

The notation suggested here yields a representation, which is applicable to serial, parallel and hybrid kinematic structures and can be applied in the same way to locomotive and humanoid robots. With the extended DH-parameters we are able to describe spherical, cardan, cylindrical, revolute, and prismatic joints following the well-known intuitive notation by Denavit and Hartenberg. When applying the proposed notation, it is possible to describe spherical or cardan joints with only 5 parameters instead of $3x4$ parameters or respectively $2x4$ parameters necessary when using the traditional DH-notation. With the suggested graph representation we can refer to any element of any kind of kinematic structure uniquely. The notation suggested here has been used in a kinematic simulation system, able to deal with various types of kinematic structures. For storing further parameters, e. g., inertial parameters, only the node attributes of the graph representation need to be augmented. We suggest such a unified notation in order to bring the community to one single notation for an easier way to exchange tools and ideas where closed kinematic chains are involved.

Acknowledgements. We are grateful to the German Research Foundation (DFG) for supporting the Collaborative Research Center (SFB) 562.

References

1. Gough, V., Whitehall, S.: Universal tyre test machine. In: Proc. of the IX Int. Techn. Congr. F.I.S.I.T.A. (1962)
2. Stewart, D.: A platform with 6 degrees of freedom. In: Proc. of the Institution of Mechanical Engineers, pp. 371–386 (1965)
3. Hesselbach, J., Frindt, M.: Structural classification and systematic design of machines basing on parallel structures. In: Proc. of the International Symposium on Robotics, pp. 65–70 (2000)

4. Clavel, R.: Delta, a fast robot with parallel geometry. In: Proc. of International Symposium on Idustrial Robots (ISIR), pp. 91–100 (1988)
5. Miller, K.: The proposal of a new model of direct drive robot delta-4 dynamics. In: Proc. of the IEEE International Conference on Advanced Robotics, Tokyo, Japan, pp. 411–416 (1992)
6. Gosselin, C.M., Angeles, J.: Singularity analysis of closed-loop kinematic chains. IEEE Trans. on Robotics and Automation 6(3), 281–290 (1990)
7. Merlet, J.-P.: Parallel Robots. Kluwer, Dordrecht (2000)
8. Pierrot, F., Marquet, F., Company, O., Gil, T.: H4 parallel robot: Modeling design and preliminary experiments. In: Proc. of the IEEE International Conference on Robotics and Automation, Seoul, Korea, pp. 3256–3261 (2001)
9. Gosselin, C.M., Masouleh, M.T., Duchaine, V., Richard, P.-L., Foucault, S., Kong, X.: Parallel mechanisms of the multipteron family: Kinematic architectures and benchmarking. In: Proc. of the IEEE International Conference on Robotics and Automation, pp. 555–560 (2007)
10. Kong, X., Gosselin, C.M.: Kinematics and singularity analysis of a novel type of 3-crr 3-dof translational parallel manipulator. The International Journal of Robotics Research 21(9), 791–798 (2002)
11. Merlet, J.-P.: Solving the forward kinematic of a gough-type parallel manipulator with interval analysis. The International Journal of Robotics Research 23(3), 221–235 (2004)
12. Thomas, U., Maciuszek, I., Wahl, F.M.: A unified notation for serial, parallel and hybrid kinematic structures. In: Proc. of the IEEE International Conference on Robotics and Automation, pp. 2868–2873 (2002)
13. Denavit, J., Hartenberg, R.S.: A kinematic notation for lower pair mechanisms based on matrices. ASME Journal of Applied Mechanics 22, 215–221 (1955)
14. Mayer, B.S.-O., Gosselin, C.: Singularity analysis and representation of the general gough-stewart platform. The International Journal of Robotics Research 19(3), 271–288 (2000)
15. Kim, W.K., Byun, K., Cho, H.S.: Closed-forward position solution for a 6-dof 3-ppsp parallel mechanism and its implementation. The International Journal of Robotics Research 20(1), 85–99 (2001)
16. Pio Belifiore, A.D.B.N.: Connectivity and redundancy in spatial robots. The International Journal of Robotics Research 19(12), 1245–1261 (2000)
17. Erdmann, A.G., Sandor, G.N.: Mechanism Design – Analysis and Synthesis. Prentice-Hall, Englewood Cliffs (1991)
18. Tsai, L.W.: The kinematics of spatial robotic bevel-ear trains. IEEE Journal of Robotics and Automation 4(2), 150–156 (1988)
19. Veitschegger, W.K., Wu, C.: A method for calibrating and compensating robot kinematic errors. In: Proc. of the IEEE International Conference on Robotics and Automation (1987)

Structure and Type Synthesis of Parallel Manipulators

Matthias Frindt, Mathias Krefft, and Jürgen Hesselbach

Abstract. Parallel robots have received increasing attention from researchers and developing engineers over the past decade, due to their inherent advantages in terms of velocity, stiffness and accuracy. However, choosing the best mechanism with optimal dimensions for best performance is a challenging and complex task. Depending on the application, different performance criteria have to be taken into account following contradictory aims. The main idea of this paper is to present the fundamentals for structure synthesis as well as a power-based optimization of parallel robots according to given application requirements, with the focus on the dynamic performance for high speed assembly tasks.

1 Introduction

It is well known that parallel robots have several significant advantages compared to their serial counterparts. Yet up to now, very few of these machines have been commercially realized, e.g., the ABB FLEXPICKER and the Adept QUATTRO, many sophisticated machines are still research objects. The reasons for this lack of technological transfer can ascribed to in structural drawbacks of parallel robots, like the limited workspace and movability. For a better exploitation of parallel robots they have to be optimized in several ways. In addition to improvements in the field of sensor-based robot motion control and sophisticated new machine parts and components, structure synthesis and optimization are the most important challenges for the developing engineer. Thus, in literature one will find various parallel structures. Still, there is no comprehensive systematic classification and evaluation scheme of parallel structures. The kinematic features of parallel robots strongly depend on their

Matthias Frindt · Mathias Krefft · Jürgen Hesselbach
Technische Universität Braunschweig, Institute of Production Automation and Machine Tools,
Langer Kamp 19b, 38106 Braunschweig, Germany
e-mail: {m.frindt,m.krefft,j.hesselbach}@tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 17–37.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

mechanical design, their arrangement of the joints within the chains, and the geometrical parameters. The starting point for a generic structure synthesis is a systematic classification of parallel robots in consideration of the characteristic properties. In this paper we firstly stress the task of finding an appropriate mechanical structure for the given requirements. After presenting a general classification of parallel structures [1, 2], we present strategies to develop and extend new mechanisms.

In the next step we determine the best geometrical parameters of the parallel structure for the best kinematic and dynamic properties. In general, most optimization strategies currently utilized consider the kinematics and dynamics of the mechanism in a successive manner [3, 4]. Moreover, the dynamic properties of a parallel robot are optimized after the kinematics are defined. The main focus of these approaches is the optimization of two features: the workspace and the Jacobian matrix. This is a typical approach, since the tools for kinematic and dynamic analysis are often not the same, and the dynamic analysis is usually very difficult to perform.

In the last years, the dynamics of parallel mechanisms have been addressed by an increasing number of authors. Di Gregorio *et al.* [5] define the swiftness of a manipulator, and Zhang *et al.* optimize a parallel kinematic machine tool with three DOF under kinetostatic aspects [6]. However, it should be pointed out that these approaches focus only on one criterion. Yet, the consideration of contradictory design goals is a crucial point in the optimization/development of parallel robots. In this paper we will discuss different performance criteria. Our goal is to modify them with respect to their physical meaning and with respect to the underlying application requirements. By integrating the kinematics and dynamics in the design process, enhanced parallel robots can be developed.

2 Systematics of Machines Basing on Parallel Structures

2.1 Classification

With a definition of the Theory of Mechanisms the design process of parallel structures starts by defining the kinematic chain of the mechanism. The kinematic chain consists of quasi-rigid links which are connected by joints (kinematics pairs) and allow relative motion of the neighboring links. This chain defines the structural design of the mechanism by determining how the different links are connected and how the DOF is distributed among the joints within the mechanism. The types of the single joints and the geometrical dimensions of the links are not defined by the kinematic chain. It must be distinguished between open and closed kinematic chains (Fig. 1) [7, 8]. In general, the joints of a kinematic chain are emblematized by circles. For spatial chains it is necessary to denote the DOF f for each single joint.

According to this definition a parallel structure can be classified as a mechanism with one (or more) closed kinematic chain(s): The end-effector (working platform) is connected to the base platform by several guiding chains [2].

Consequently, the structural characteristics of a parallel structure are given by the number of guiding chains, their architecture and their arrangement within the

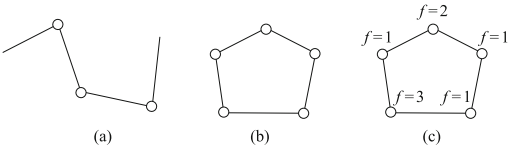


Fig. 1. Kinematic chains: (a) planar open chain (b) planar closed chain (c) spatial closed chain.

mechanism. Therefore, the first step for describing the structural design is given by the mechanical scheme, which contains information about the number of chains and their DOF [9, 10]. Figure 2 shows the mechanical scheme 6-6-6-6-6-6 and some resulting mechanical structures of known parallel structures.

By choosing the specific joints within the guiding chains and by defining the active joints, the mechanical concept of the parallel structure is obtained. Figure 2 shows three examples which can be derived from the 6-6-6-6-6-6 scheme. Each of these concepts is the starting point for the definition of mechanical structures, which are realized by determining certain constraints between the geometrical parameters. One of the examples is given by the mechanical structures according to [11, 12] (Fig. 2).

Both are based upon the same mechanical concept, but they have different kinematic characteristics according to the arrangement of their guiding chains: The HEXA structure [12] consists of three pairs of guiding chains which are arranged

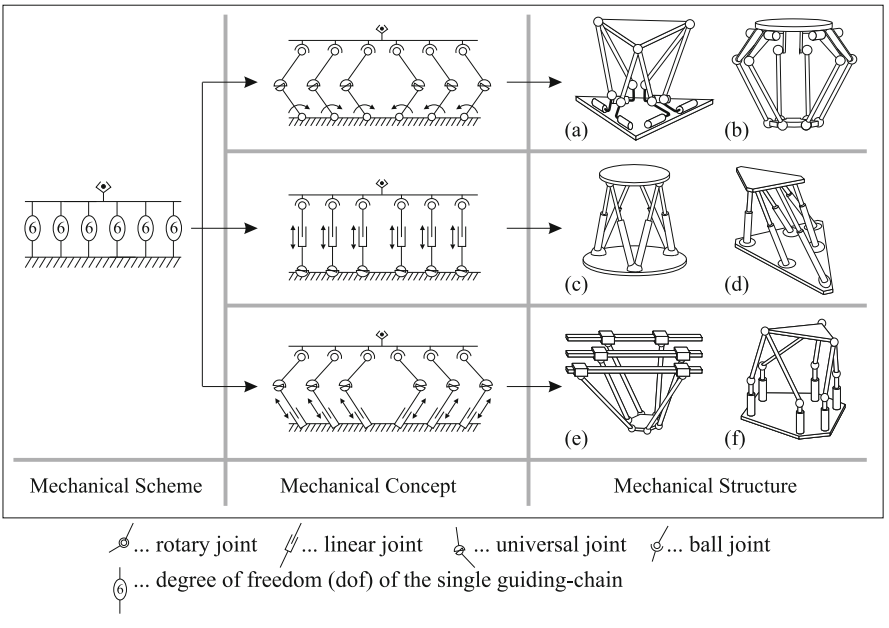


Fig. 2. Parallel structures developed from the 6-6-6-6-6-6 scheme:
(a) Hunt [11]; (b) Pierrot *et al.* [12]; (c) Gough *et al.* [13]; (d) Griffis *et al.* [14]; (e) Hebsacker [15]; (f) Merlet *et al.* [16].

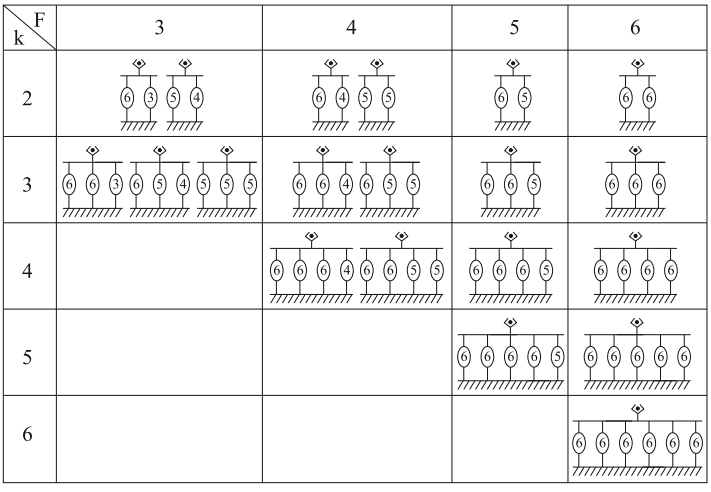


Fig. 3. Distribution of joint DOF for basic parallel structures.

symmetrically between the base platform and the end-effector. In contrast, the structure according to [11] is built in a way such the end-effector is guided by three points only. Each of these points connects the end-effector with two guiding chains to the base platform. As a result of these constraints, singularities con be avoided as long as the end-effector is moved in parallel to the base platform ($\psi = \vartheta = \varphi = 0°$).

2.2 Basic Mechanical Schemes

First, the focus is set on a very special group of parallel structures which can be derived from so-called basic mechanical schemes [10]. These are characterized by non-branched guiding chains connecting the base platform with the end-effector. Furthermore, the number of guiding chains of basic mechanical schemes is not greater than the DOF of the whole mechanism. Applying this condition, the Grübler or Kutzbach formula for calculating the DOF of spatial mechanisms can be derived for basic mechanical schemes [1]:

$$\sum_i f_i = F + 6\,(k - 1)\,, \tag{1}$$

where k denotes the number of guiding chains, $\sum f_i$ the sum of joint DOF and F the DOF of the mechanism itself. Using Equation (1), the sums $\sum f_i$ are given in Fig. 3 in dependency of the DOF F and the chain number k. In addition, Fig. 3 shows how these sums of the joint DOF can be distributed on each guiding chain [17]. Thus, a basic mechanical scheme with $F = 3$ and three guiding chains must possess exactly a joint DOF of 15, for instance. These joint DOF can be distributed on the chains in the following manner: 663, 654 or 555.

2.3 Design of the Guiding Chains

The mechanical concept of a parallel structure (see Fig. 2) is mainly defined by the design of the guiding chains. In order to use the potential of structures with parallel structure, some simple rules must be taken into account concerning the design of the guiding chains [10]:

All drives must be arranged close to — or better in — the fixed frame in order to reduce the moved masses of the structure. If necessary, this can be done by using planar sub-chains inside of multi-driven guiding chains (e.g., Fig. 10). Non-actuated (passive) prismatic joints should not be used because of sticking effects. The number of joints should be as low as possible in order to increase the rigidity (≤ 3 for each chain). Additionally, branches can be used to avoid bending forces in the rods and to improve the stiffness of the structure (e.g., Fig. 9).

Figure 4 shows a selection of typical guiding chains with $\sum f_i = 6$. In [10], these guiding chains are called characteristic guiding chains as they determine the characteristic geometrical and kinematical features such as the shape and size of the workspace and the transmission behavior of the mechanism. Parallel structures which are based on the same characteristic guiding chains can be combined to the same class of mechanisms.

An important criterion for the selection of the correct characteristic guiding chain is its range of movement. It can be used to get a first rough estimation of the working space of the complete mechanism. For example, the characteristic guiding chain D_3D_2D is taken into account. Figure 5 shows the range of movement of this guiding chain.

The guiding chain D_3D_2D is well known from the so-called HEXA structure (Fig. 2(b)). Figure 6 shows sectional views of the ranges of movement of the six guiding chains of the HEXA structure. The resulting workspace of the structure (orientation of the end-effector: $\psi = \vartheta = \varphi = 0$) is highlighted by shading. In [10] a more detailed overview on the specific attributes of several characteristic guiding chains is given.

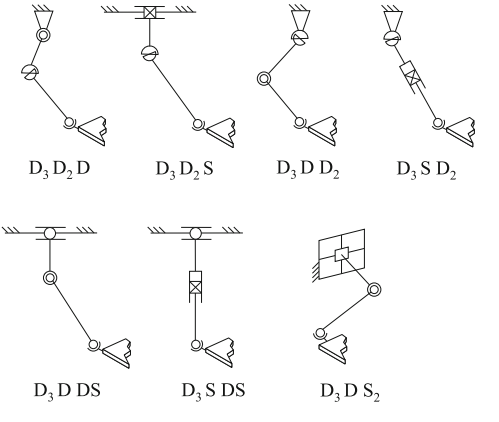


Fig. 4. Characteristic guiding chains with $\sum f_i = 6$.

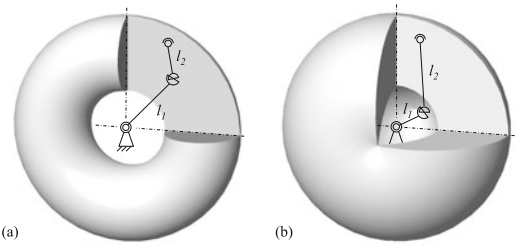


Fig. 5. Range of movement of guiding chain D_3D_2D: (a) $l_1 > l_2$ (b) $l_1 < l_2$.

Particular attention must be paid to the design of the guiding chains with $\sum f_i < 6$, since they must reduce the DOF F of the moving platform without limiting the workspace. In general, the parallel architecture implies that all movements of the end-effector are totally coupled resulting in constrained movements for mechanisms of 3 to 5 DOF. In [18], the idea of parallel structures of the PENTA class has been presented. These structures allow the end-effector to be positioned in x-, y- and z-direction and to be oriented according to x- and y-axis. In addition, the structural design leads to an accompanying orientation relative to the z-axis, which is the axis of the end-effector (tool). For machining tasks using rotating-tools, this accompanying orientation has no disturbing consequences. Compared to machines of the HEXA class one drive can be omitted reducing the costs of the machine. Figure 7 shows a PENTA structure with a passive 5-DOF guiding chain D_2SD_2. The workspace and the transmission behavior are determined by the characteristic D_3D_2D guiding chain (Fig. 4 and 5). Moreover, an additional constraint condition has to be taken into account [18, 19]. Similar examples for parallel mechanisms with reduced DOF ($F = 3 \ldots 4$) are discussed in [20, 21].

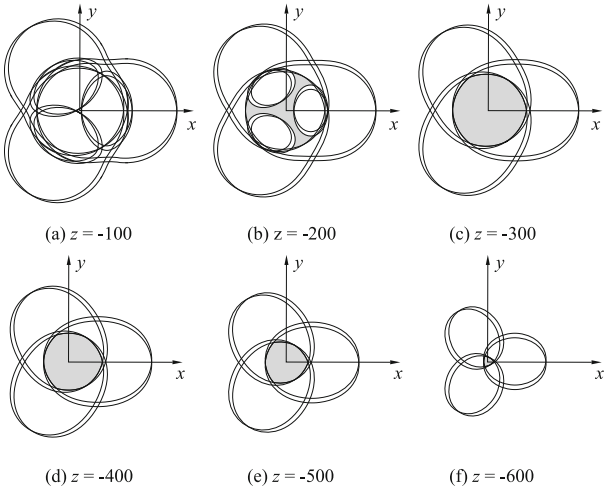


Fig. 6. Workspace of the HEXA structure derived from the analysis of the guiding chains ($\psi = \vartheta = \varphi = 0°$).

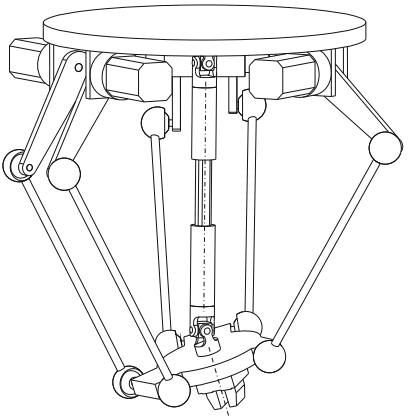


Fig. 7. PENTA structure of the D_3D_2D class (patent pending [22]).

2.4 Expansion of Additional Branches

Basic mechanical schemes as introduced in Sec. 2.2 can be expanded by additional branches. This strategy can be used to adapt the mechanism to the given application by locking single movements of the end-effector or by increasing the rigidity of the mechanism. Furthermore, additional branches can be used for fixing all drives in the frame when the number of guiding chains is lower than the number of drives. In [1] it is shown that each expansion of a spatial structure needs six joint DOF if the structure's DOF F is not to be changed. According to Fig. 8, the additional branch can either be added within a guiding chain (b) or as an additional chain between the base platform and the end-effector (a).

One of the most popular examples for such an expansion is given by Clavel's famous Delta structure [23]. According to Fig. 9, the Delta structure can be derived by expanding a basic mechanical scheme (555) with three guiding chains, each containing one rotary and two universal joints (a). Every chain is expanded by an additional branch raising the number of joint DOF to eleven (b). In order to improve the force transmission the remaining universal joints are replaced by ball joints (c) resulting in six identical DOF altogether that have no influence on the movement of the end-effector (rotation of the rods among their main axis). The use of ball joints guarantees that all the rods are free of bending forces. As a result of the expansion the parallelograms formed by the additional branches lock the rotational movements of the end-effector, meaning that the Delta structure exhibits pure translational movement in x-, y- and z-direction (Fig. 9).

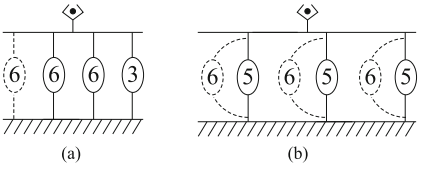


Fig. 8. Expansion by additional branches.
(a) Additional guiding chain
(b) Branch within guiding chain.

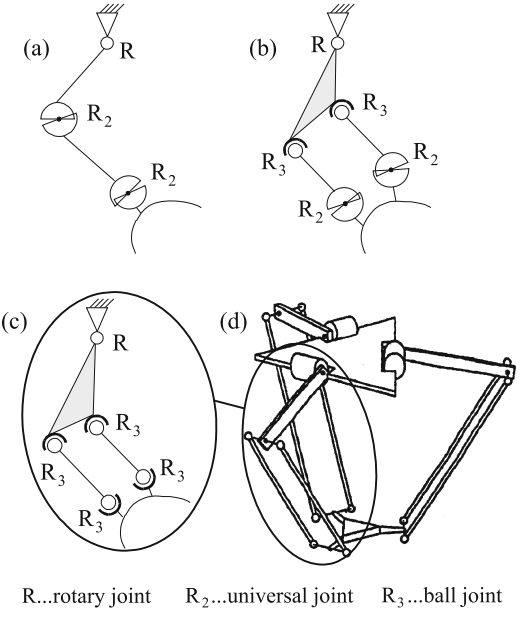


Fig. 9. Development of the Delta structure: (a) Guiding chain with six joint DOF (b) Expansion by additional branch (+ six joint DOF) (c) Replacing the universal joints by ball joints (d) Delta structure.

The idea of the Delta can also be applied on structures with linear drives [24]. Another example for an extended basic structure with improved force transmission is given in [25].

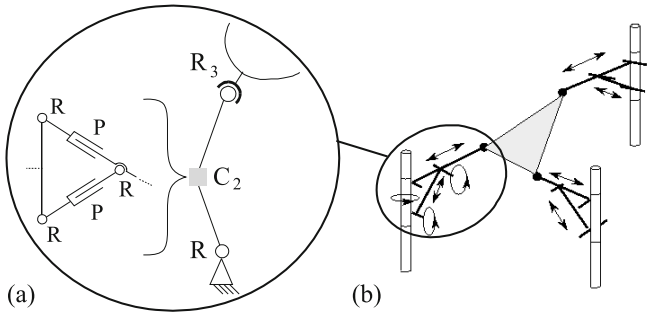
2.5 Expansion of Planar Sub-chains

The extension of planar sub-chains is characterized as an extension of additional branches. One example for a parallel structure with planar sub-chains is given by the Stewart Platform [26] in Fig. 10. This structure has six DOF and contains three guiding chains between the base platform and the end-effector. According to Fig. 3, the sum of the joint DOF must equal 18 leading to a 666 distribution. As shown in Fig. 10(a), the planar sub-chain must be considered separately when counting the DOF of one single guiding chain. Virtually, this planar sub-chain is replaced by a complex joint C. The DOF of this joint is given by the Grübler formula for planar mechanisms [9]:

$$F_C = 3(n-1) - 3g + \sum f_i, \tag{2}$$

where n is the number of links, g the number of joints, and $\sum f_i$ the sum of joint DOF. From $n = 5$, $g = 5$, and $f_i = 1$ proceeds $F_C = 2$ for the complex joint C of the Stewart Platform. Consequently every guiding chain contains six joint DOF altogether.

Using this extension, both drives of each guiding chain can be arranged near the frame, which results in reduced inertia. Similar examples are given in [27, 28].



R...rotary joint P...prismatic joint R_3...ball joint C_2...complex planar joint

Fig. 10. Architecture of the Stewart Platform - (a) Guiding chain containing planar sub-chain (b) Stewart Platform [26].

3 Performance Criteria

The most important requirements of parallel robots are the workspace, velocity, accuracy (stiffness), and forces. But these keywords must be examined at more precisely and put into mathematical expressions. The characteristic functions allow the establishment of performance criteria and help to determine the degree of fulfillment of a parallel robot structure with regard to the requirements. First, it must be noted again that all requirements and developed characteristic functions assigned to parallel robot structures are generally neither constant nor isotropic but they depend on the location or pose (position and orientation) of the end-effector.

3.1 Workspace

The workspace is one of the most important characteristics of parallel robots to be analyzed. This analysis of the workspace is a difficult problem, since the workspace boundary hyper planes are defined by a set of highly nonlinear equations. Indeed, in the most cases analytical calculation methods are not easy to perform and discretizing methods may be helpful. Additionally, they are not very efficient for standardized optimization approaches because they depend on the type of the structure. Thus, a numeric approach for calculation of the workspace volume is used. The main idea is the discretization of the workspace in F dimensions. Considering the application requirements, we distinguish between the position workspace and the orientation range, which is a condition to be fulfilled at each position inside the workspace. In contrast to the volume, the shape of the workspace is a feature that is neglected in the literature. Some authors approximate the workspace of a parallel robot with a cuboid, which can be evaluated by the extreme reaches along the x-, y- and z-axis [4]. Considering the position workspace, we can see that an outside approximation is unsuitable, because a large part of the space is not covered by

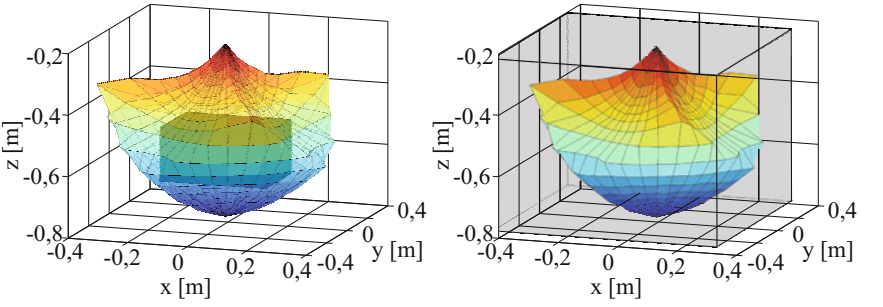


Fig. 11. Workspace: inside and outside approximation.

the workspace of the parallel robot. Therefore, we have developed an algorithm for finding the maximum cube/cuboid inside a parallel robots workspace (Fig. 11).

We have to bear in mind that the calculation of the workspace $\mathcal{W}$ is subject to some important constraints. First of all, we must consider the orientation of the end-effector, limitation of the actuated joints, i.e. the drives, as well as the passive joints. Additionally, points with collisions of movable parts are excluded. Thus, a key issue of designing parallel robots is to consider fixed value requirements R_i inside the optimization approach. We will make a mistake if these properties are modeled as properties to be optimized, that means to be maximized or minimized, e.g., in a multi-criteria optimization approach, we cannot guarantee that there will be no singularities if we maximize the determinant of the Jacobian. Thus, the application workspace $\mathcal{W}^*_{task}$ is the sum of all points X_i where the properties P_i of the parallel robot fullfil the fixed application requirements R_i. As mentioned above, the necessary orientation ability of the working platform in each point is also modeled as a constraint. A reasonable performance criterion is the workspace to footprint ratio:

$$\eta_{\mathcal{W}} = \frac{V\left(\mathcal{W}^*_{task} = \{X_i | P_i(X_i) < R_i\}\right)}{A_{footprint}}. \tag{3}$$

3.2 Velocity Transmission

The kinematic and dynamic analysis of parallel robots is strongly associated with the linear transmission of the drive velocities by the Jacobian matrix $\mathbf{J}_a$:

$$\dot{\mathbf{X}} = \mathbf{J}_a \dot{\mathbf{q}}. \tag{4}$$

In the literature, a large number of approaches discussing this transmission behavior can be found [29, 30]. Thus, the standardized drive velocities $\dot{\mathbf{q}}_i$, which describe the surface of a F-dimensional hyper sphere in the joint space, are transmitted on a hyper ellipsoid in the Cartesian space representing the time derivates of the end-effector pose. This transmission is characterized by the extension/amplification and deformation of the hyper sphere and the phase shifting of the output vector

$\dot{\mathbf{X}}$ according to the input vector $\dot{\mathbf{q}}$. The maximum amplification of an input (drive-) velocity is given by the spectral norm of the Jacobian matrix:

$$\|\mathbf{J}_a\|_2 = \max_{\|\dot{\mathbf{q}}\|_2 \neq 0} \left(\frac{\|\mathbf{J}_a \dot{\mathbf{q}}\|_2}{\|\dot{\mathbf{q}}\|_2} \right). \tag{5}$$

In the same way, $\left\|\mathbf{J}_a^{-1}\right\|_2^{-1}$ gives the minimum amplification. The spectral norm can be calculated with the square root of the maximum eigenvalue $\xi_{\max}$ of $\mathbf{J}_a\mathbf{J}_a^T$. Since $\mathbf{J}_a\mathbf{J}_a^T$ and $\mathbf{J}_a^T\mathbf{J}_a$are equal, the singular value $\sigma_{\max}$ is given by

$$\sigma_{\max}(\mathbf{J}_a) \equiv \|\mathbf{J}_a\|_2 = \sqrt{\xi_{\max}(\mathbf{J}_a^T\mathbf{J}_a)}. \tag{6}$$

It is the maximum semi half axis of the hyper ellipsoid, and

$$\sigma_{\min}(\mathbf{J}_a) \equiv \sqrt{\xi_{\min}(\mathbf{J}_a^T\mathbf{J}_a)} = \sigma_{\max}^{-1}\left(\mathbf{J}_a^{-1}\right) \tag{7}$$

is the minimum semi half axis. Thus, the deformation of the hyper ellipsoid can be described by the condition number κ of the Jacobian matrix:

$$\kappa(\mathbf{J}_A) = \frac{\sigma_{\max}(\mathbf{J}_a)}{\sigma_{\min}(\mathbf{J}_a)} = \|\mathbf{J}_a\|_2 \left\|\mathbf{J}_a^{-1}\right\|_2. \tag{8}$$

Using the inverse of the condition number the range of values can be shifted from $[1, \infty]$ to the interval $[0, 1]$ cf. [31]. Even though, the inverse condition number is a local value, which depends on the pose of the end-effector, many authors, e.g., Angeles [30], use this value for the optimization of parallel robots.

The aim of this approach is a uniform velocity transmission in all directions. If $\kappa^{-1} = 1$, the structure is called *isotropic*, irrespective of other poses [30]. In addition, the general opinion that $\kappa^{-1} = 1$ means the robot can reach the same velocities in all directions is wrong from a physical point of view. This fact has serious consequences for the use of the singular values. Figure 12 shows the Biglide structure based on the PRRRP kinematic chain in a kinematic isotropic position (P=prismatic joint, R=rotational joint). However, considering the maximum velocities of each drive, the maximum and minimum end-effector velocity differs as depicted in Tab. 1. The reason for this lies in the fact that the input vector $\dot{\mathbf{q}}$ has to be described by a hyper square ($\text{index}_\square$) with a side length of $2q_{\max}$.

If $q_{1,max} = q_{2,max}$, the ratio between the maximum the minimum Cartesian velocity is $\kappa_v = \left\|\dot{\mathbf{X}}_{\max}\right\|_2 \left\|\dot{\mathbf{X}}_{\min}\right\|_2^{-1} = 1,41$, which is the same as the ratio between the input drive velocities $\kappa_q = \|\mathbf{q}_{\max}\|_2 \|\mathbf{q}_{\min}\|_2^{-1} = 1,41 = \kappa_v$, thus $\kappa = \kappa_v/\kappa_q = 1$. Table 1 shows the spectral norm of the velocities and conditions for two positions of the PRRRP-structure as shown in Fig. 12. Thus, the extreme Cartesian velocities can be calculated by

$$\dot{\mathbf{X}}_{\max(\min)} = \max(\min)\left(\|\mathbf{J}_a \dot{\mathbf{q}}_\square\|_2\right) \tag{9}$$

with the drive velocities

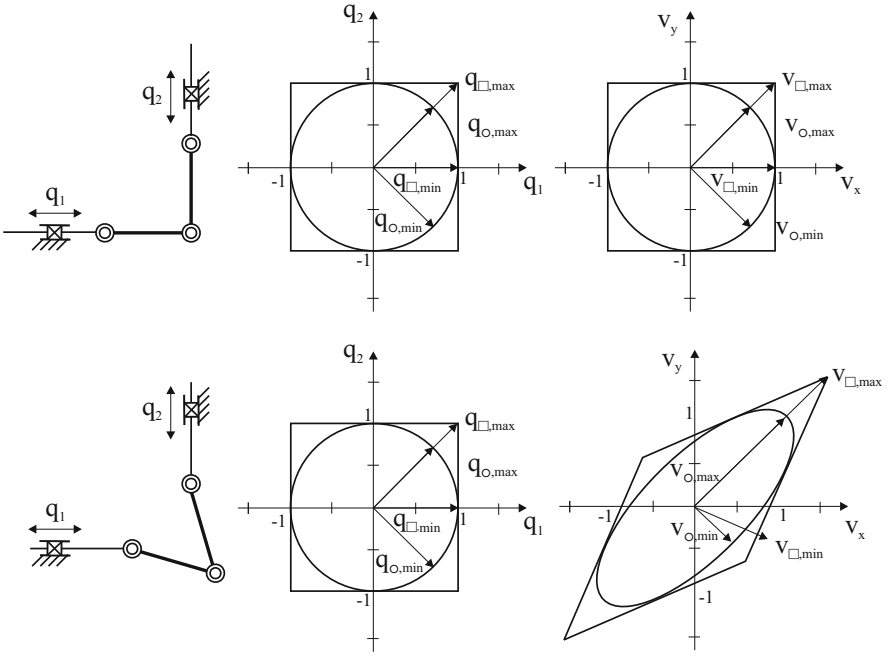


Fig. 12. Velocity transmission of the PRRRP-structure.

Table 1. Velocity transmission of the PRRRP structure.

Input	$q_{\min}$	$q_{\max}$	$v_{\max}$	$v_{\min}$	κ_q	κ_v	κ
1: sphere	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1: square	1.00	1.41	1.00	1.41	1.41	1.41	1.00
2: sphere	1.00	1.00	0.70	1.77	1.00	2.55	2.55
2: square	1.00	1.41	0.92	2.51	1.41	2.74	2.40

$$\dot{\mathbf{q}}_{\square} = \left[\mu_1 q_{1,\max(\min)} \ \cdots \ \mu_F q_{F,\max(\min)} \right] \tag{10}$$

and the parameters

$$\mu_i \in [-1, 1] \wedge \mu_j \in \{-1, 1\}, i = \{1, \ldots, F\} \setminus \{j\}. \tag{11}$$

The first example shows one of the most common problems . Furthermore, if we analyze a parallel robot with $F = 6$, we will generally mix different physical units:

Table 2. Singular values and direction for different approaches.

Type	Value	→ x	→ y	→ z	↻ x	↻ y	↻ z
Standardized Jacobian							
σ_{max}	0.44	0.00	-0.36	0.00	0.28	0.00	0.04
σ_{min}	0.19	-0.02	0.00	-0.10	0.00	0.05	0.00
Separated Jacobian							
v_{max}	0.35	0.11	-0.34	0.00	7.68	2.49	0.00
Ω_{max}	9.11	-0.21	-0.23	0.00	6.75	-6.12	0.00
Pure velocities							
v_{max}	0.17	0.00	0.00	0.17	0.00	0.00	0.04
Ω_{max}	-4.47	0.00	0.00	0.00	0.00	0.00	-4.47

$$[\mathbf{J}_a] = \begin{cases} \begin{bmatrix} 1 & 1 \\ rad/m & rad/m \\ m/rad & m/rad \\ 1 & 1 \end{bmatrix} & \forall\ \underline{\mathrm{P}}\mathrm{US}, \mathrm{U}\underline{\mathrm{P}}\mathrm{S} \\ & \\ & \forall\ \underline{\mathrm{R}}\mathrm{US} \end{cases} . \tag{12}$$

In this case, each performance criterion which is based on the Jacobian matrix has no physical meaning. In addition, the values of the two submatrices have different magnitudes, and one submatrix will dominate the other one. Thus some authors such as Angeles [30] define a constant characteristic length arbitrarily. Kirchner [32] uses the projection of the platform vectors on the orientation planes. But for all that, the use of the standardized matrix is problematic, as shown in Tab. 2. The table gives the six singular values of a HEXAPOD structure (6-UPS) for a given pose. As can be seen, the maximum singular values describe a combined motion, i.e. a translation in y-direction and a rotation around the x-axis.

The following approach using two criteria for translation and orientation will avoid this problem. The main idea is to split the Jacobian matrix in two submatrices containing the translational (index t) and rotational (index t) transmission behavior:

$$\mathbf{J}_a = \left[\mathbf{J}_{a,t}^T \, \mathbf{J}_{a,r}^T \right]^T . \tag{13}$$

Now Eq. 4 can be separated as follows:

$$\mathbf{v} = \mathbf{J}_{a,t}\dot{\mathbf{q}}, \mathbf{J}_{a,t} \in \mathbb{R}^{3\times 6}, \ \boldsymbol{\Omega} = \mathbf{J}_{a,r}\dot{\mathbf{q}}, \mathbf{J}_{a,r} \in \mathbb{R}^{3\times 6}. \tag{14}$$

These two equations will give the extreme translational and rotational velocities

$$\mathbf{v}_{\max(\min)} = \max(\min)\left(\left\| \mathbf{J}_{a,t}\tilde{\mathbf{V}}_{\dot{\mathbf{q}}} \right\|_2 \right) \tag{15}$$

and

$$\boldsymbol{\Omega}_{\max(\min)} = \max(\min)\left(\left\| \mathbf{J}_{a,r}\tilde{\mathbf{V}}_{\dot{\mathbf{q}}} \right\|_2 \right) \tag{16}$$

as shown in Tab. 2. The corresponding transmission ratios are $\xi_t = \mathbf{v}_{\min}/\mathbf{v}_{\max}$ and $\xi_r = \mathbf{\Omega}_{\min}/\mathbf{\Omega}_{\max}$. These velocities are clearly no pure velocities and the corresponding eigenvectors will lead to motion in all directions. This means v_{max} describes the maximum translational velocity with changing orientation of the end-effector. This performance criterion fits the maximum speed criterion of common industrial robots, e.g., the maximum speed of a SCARA robot in a configuration, in which both arms are elongated and, therefore the orientation cannot remain constant. The calculation of pure velocity criteria is as follows. The starting point is the calculation of the inverse submatrices

$$\mathbf{J}_a^{-1} = \left[\mathbf{J}_{a,t}^{inv} \; \mathbf{J}_{a,r}^{inv} \right], \mathbf{J}_{a,t}^{inv}, \mathbf{J}_{a,r}^{inv} \in \mathbb{R}^{6x3}. \tag{17}$$

Then the corresponding Cartesian eigenvectors $\tilde{\mathbf{V}}_{\mathbf{v}}$ and $\tilde{\mathbf{V}}_{\mathbf{\Omega}}$ of $\mathbf{J}_{a,t}^{inv}$, and $\mathbf{J}_{a,r}^{inv}$ must be calculated.

With the linear transmission $\mathbf{V}_{\dot{\mathbf{q}}} = \mathbf{J}_a^{-1} \left[\tilde{\mathbf{V}}_{\mathbf{v}}^T \; \mathbf{O} \right]$ and $\mathbf{V}_{\dot{\mathbf{q}}} = \mathbf{J}_a^{-1} \left[\tilde{\mathbf{V}}_{\mathbf{v}}^T \; \mathbf{O} \right]$ respectively, with $\mathbf{O} \in \mathbb{R}^{3x3}$, the eigenvectors are transformed in the joint space. Finally, the pure translations are $\left[\left\| \mathbf{v}_{\dot{\mathbf{q}}_1} \right\|_2^{-1} \left\| \mathbf{v}_{\dot{\mathbf{q}}_2} \right\|_2^{-1} \left\| \mathbf{v}_{\dot{\mathbf{q}}_3} \right\|_2^{-1} \right]$ and $\left[\left\| \mathbf{v}_{\dot{\mathbf{q}}_4} \right\|_2^{-1} \left\| \mathbf{v}_{\dot{\mathbf{q}}_5} \right\|_2^{-1} \left\| \mathbf{v}_{\dot{\mathbf{q}}_6} \right\|_2^{-1} \right]$. The advantage of this approach is the ability to deal with a physically correct and clear description of the velocity as shown in Tab. 2.

3.3 Forces and Dynamic Effects

For high speed applications, the most important aim is to increase the end effector velocity. Therefore, most authors maximize the velocity transmission as shown in Sec. 2.2. Nevertheless, this approach has the risk of a saturation of the drive force, since velocity and force transmission are inverse. Thus, the analysis of the drive forces τ has to be included in the optimization approach. However, it becomes difficult to assign a physical interpretation when authors such as Kirchner [32] and Ottaviano [4] mix forces and torques. Their approach yields physically inconsistent results when different units are mixed. An appropriate approach which is made for serial robots by Bowling et al. [33] is well suited for our task. The main idea is to separate the different force components, e.g., inertia, torques, forces, etc.

$$\tau = \tau_{\mathbf{f}_L} + \tau_{\mathbf{m}_L} + \tau_{\dot{\mathbf{v}}} + \tau_{\dot{\Omega}}. \tag{18}$$

The addition of each contribution can be represented geometrically by mapping one force ellipsoid onto every point on the surface of the other as shown in Fig. 13.

To simplify the problem, we firstly consider the external loads and inertia effects. The separation of the geometric Jacobian

$$\mathbf{J}_g^T = \left[\mathbf{J}_{g,t}^T \; \mathbf{J}_{g,r}^T \right], \tag{19}$$

which differs from the analytic Jacobian $\mathbf{J}_a$ for robots with $F > 3$, yields

$$\tau = \mathbf{J}_{g,t}^T \mathbf{f}_L + \mathbf{J}_{g,rot}^T \mathbf{m}_L + \mathbf{B}_t \dot{\mathbf{v}} + \mathbf{B}_r \dot{\Omega}, \tag{20}$$

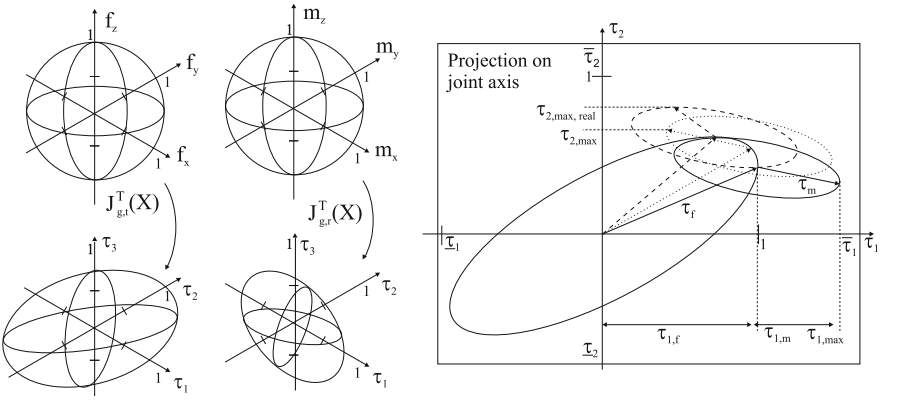


Fig. 13. Addition of the single force terms and their projected contributions in the direction of each drive.

with the inertia matrix $\mathbf{B} = \left[\mathbf{B}_t^T \ \mathbf{B}_r^T\right]^T = \mathbf{J}_{g,t}^T \sum \left(\mathbf{J}_j^T \mathbf{M}_j \mathbf{J}_j\right)$ where, $\mathbf{M}_j$ is the mass matrix of each body j. For instance the transmission ellipsoid of the external load $\mathbf{f}_l$ is given by:

$$\tau_{\mathbf{f}_L,i}^T \left(\mathbf{J}_{g,t,i}^T \mathbf{J}_{g,t,i}\right)^+ \tau_{t,i}^T = |f_L|^2 \,. \tag{21}$$

This equation can be understood by mapping a force circle into the space of the drive forces. Here, a circle is used, because it represents a maximum force vector with constant length and arbitrary direction. These mappings are valid, because we use the Moore-Penrose pseudoinverse

$$(\mathbf{J}_{g,\ldots}^T)^+ = \left(\mathbf{J}_{g,\ldots} \mathbf{J}_{g,\ldots}^T\right)^{-1} \mathbf{J}_g, \tag{22}$$

since the single contributions are of a smaller dimension than the drive force vector. The difficulty of Bowling's [33] approach is the calculation of the maximum projection of each contribution on each drive axis in the space of the drive forces. Figure 13 shows that this projection cannot be calculated by using the greatest singular values of the separated geometric Jacobian submatrices, e.g., $\mathbf{J}_{g,t}^T$ and $\mathbf{J}_{g,r}^T$. Instead of Bowling's numerical approach, we used the maximum singular values of each row to calculate the maximum projection of each contribution.

$$\mathbf{J}_g^T(\mathbf{X}) = \begin{bmatrix} \mathbf{J}_{g,t,1}^T & \mathbf{J}_{g,r,1}^T \\ \vdots & \vdots \\ \mathbf{J}_{g,t,F}^T & \mathbf{J}_{g,r,F}^T \end{bmatrix} \tag{23}$$

This process is very simple and quite effective because it gives the exact maximum force value for each drive

$$\tau_{\max} = \max_i \left(\|\mathbf{f}_L\|_2 \left\|\mathbf{J}^T_{g,t,i}\right\|_2 + \|\mathbf{m}_L\|_2 \left\|\mathbf{J}^T_{g,r,i}\right\|_2 \right). \tag{24}$$

Notice that this equation provides the possibility of determining the worst-case contributions of all terms in Eq. (18) independently.

4 Modified Evolutionary Algorithm

The mentioned performance criteria are discontinuities, discrete and nonlinear functions with constraint conditions. Thus, the optimization problem has no convexity properties and gradient methods failed where simple stochastic methods are too time consuming. In this case, evolutionary algorithms are very effective to search for the global optima without any knowledge about the optimization problem itself. In contrast to some authors who used a weighted cost function as one criterion, our approach is a Pareto based multi criteria evolutionary algorithm.

From our point of view, a combination of all performance criteria weighted with different priority factors does not have any physical meaning. Based on a *strength Pareto evolutionary algorithm* approach (SPEA) of Zitzler [34], we have developed a modified algorithm with effective selection routines including an intelligent mutation which covers stagnancy of the optimization, dynamic extension of the parameter range and new termination methods. For instance, the set of solutions of a multiobjective optimization problem consists of all individuals for which the corresponding performance criteria cannot be improved in any dimension without degradation in another. That means all individuals which are not dominated by any other are called nondominated and form the Pareto set.

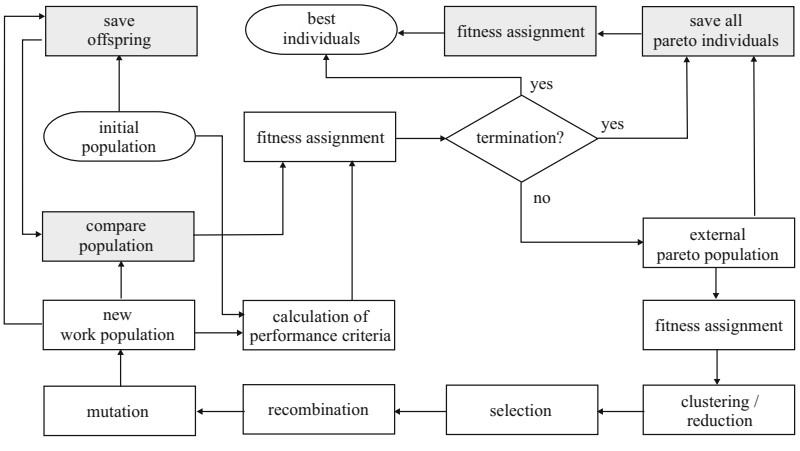


Fig. 14. Modified evolutionary algorithm.

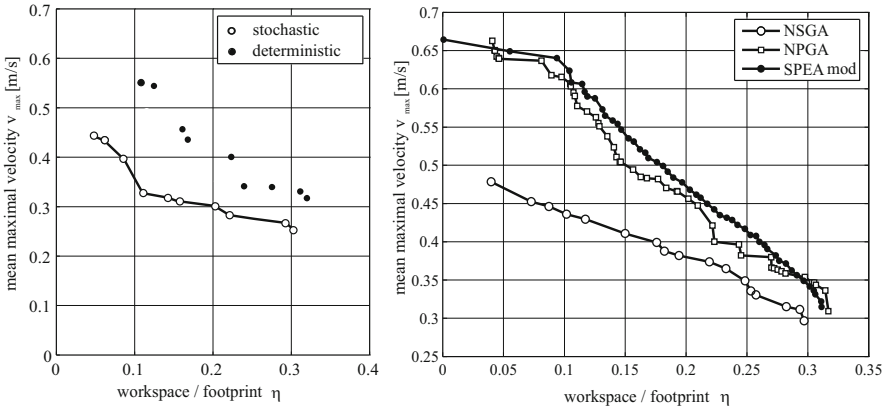


Fig. 15. Performance of the modified pareto optimization.

The main idea of the modified SPEA approach is to work with two populations. The primary population is the working population, which is the basis for the generation of new offspring, and which is created by a recombination and mutation of the individuals of the secondary population, i.e., the external Pareto archive, which contains the Pareto optimal individuals of all generations. Thus, no genetic material can get lost during the evolutionary optimization. Hence, the calculation of the performance criteria is the most time consuming task, and the performance of all offspring is stored. The used fitness assignment provides a better distribution of the solutions on the Pareto front because great importance was attached to an accurate density estimation of the individuals inside the population. Up to now, all evolutionary algorithms, e.g., the *nondominated sorting genetic algorithm* (NSGA) or the *niched Pareto genetic algorithm* (NPGA), that are used for the optimization of parallel robots suffer from an insufficient diversity since they focus only on a fast convergence [34]. Nevertheless, the modified SPEA algorithm also provides a better convergence. In contrast to standard evolutionary algorithms, we use a deterministic (instead of a stochastic population initialization) setup of the initial population (Fig. 15).

5 Simulation Results

The mechanism we have chosen for our approach is the well-known 6-DOF parallel robot HEXA (Fig. 16).

It consists of an upper and a lower platform connected by six RUS chains. The design parameters are the basis length L_A, the rod length L_{AB}, the link length L_{BC}, the platform radius L_C, and the position of the joints on the planar ($\rightarrow \beta_A = \beta_C = 0$) basis and platform, which is characterized by α_A and α_C. For symmetry reasons the parameters of all chains are equal. To increase the robot dynamics, the robot is driven by six rotational direct drives with a maximum torque of 70 Nm. The mechanism is designed for high speed handling and assembly tasks, thus, we have to

optimize the maximal translational velocity inside the workspace. A sufficient condition for this demand is given by the comparison of the applied drive torques and the inertia forces due to the necessary performance specification of an acceleration of $50\,m/s^2$ of the end-effector inside a workspace, which is also a performance parameter. Due to the higher flexibility the orientation of the platform has to be maximized as well. In addition, we have to consider the limitations of the passive joints when we analyze the workspace. In Figure 17, the properties of the Pareto front for the 100th generation run is shown. All performance criteria

$$\hat{P}_i = \frac{P_i - P_{i,\min}}{P_{i,\max} - P_{i,\min}} \tag{25}$$

are related to their maximum value during the optimization run. This standardization approach yields a better insight in the overall performance. It can be seen that

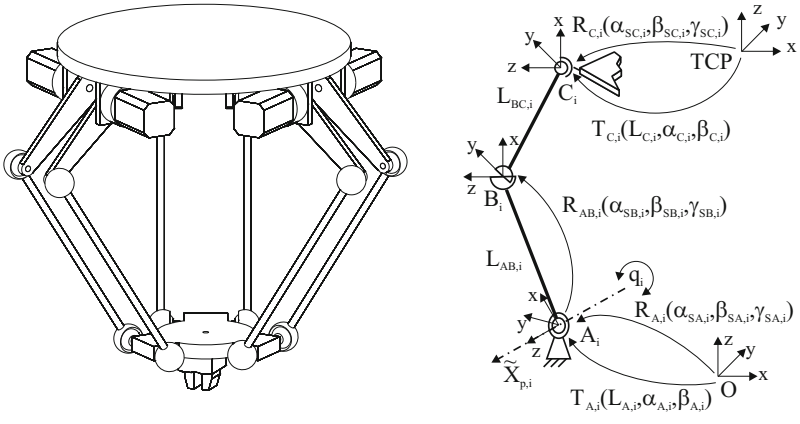


Fig. 16. HEXA: model and parameter.

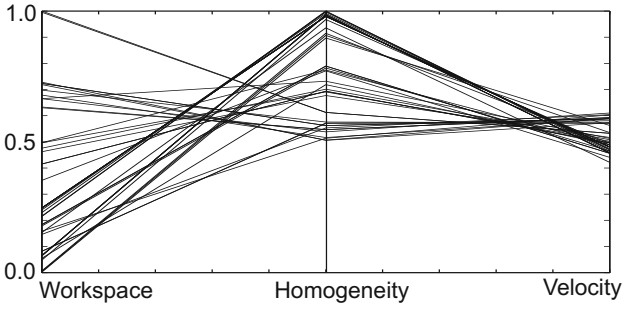


Fig. 17. Properties of all Pareto front individuals.

there are no individuals with the best values for all criteria. With regard to the applied drive power, all individuals with a standardized velocity greater than 0.61 fail the limiting force transmission requirements. Finally, choosing the best geometric set-up out of the Pareto front is the job of the design engineer: For the given requirements, the structure with $L_A = 383mm$, $L_{AB} = 265$, $L_{BC} = 587mm, L_C = 93mm$, $\alpha_A = 17.11°$, and $\alpha_C = 75.60°$ possesses the best properties.

6 Conclusion

In this paper we have presented a new concept for the development of parallel robots including structure synthesis, workspace, velocity, and forces. Starting with a systematic classification of parallel structures, we have demonstrated the evolution of several structures using different kinematic subchains with different joints. The structure optimization has been accomplished while addressing unit inhomogeneities and unifying the avoidance of fixed requirements such as collision and singularities. The analysis of requirements also yields insights into the limiting drives, and related that saturation will limit the end-effector motions and applied loads. These developments, allowing for the optimization of nonredundant parallel robots, were illustrated by means of the 6-DOF HEXA parallel robot.

References

1. Hesselbach, J., Frindt, M., Kerle, H.: Zur Struktursystematik von Parallelrobotern. Konstruktion 2(51), 36–42 (1999)
2. Kerle, H., Frindt, M., Plitea, N.: Zur Systematik von Maschinen und Geräten mit Parallelstruktur. VDI Berichte 1423, 209–224 (1998)
3. Merlet, J.-P.: Parallel robots. Solid mechanics and its applications, vol. 74. Kluwer Academic Publishers, Dordrecht (2001) ISBN 1402003854
4. Ottaviano, E., Ceccarelli, M.: Optimum design of parallel manipulators for workspace and singularity performances. In: Proceedings of the Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators, Quebec City, Canada, pp. 98–105 (2002)
5. Di-Gregorio, R., Parenti-Castelli, V.: Comparison of 3-DOF parallel manipulators based on new dynamic performance indices. In: 11th World Congress in Mechanism and Machine Science, Tianjin, China, pp. 1684–1688 (2004)
6. Zhang, D., Xi, B., Surgenor, B., Machefske, C.K.: Kinetostatic analysis of a 6-DOF parallel mechanism. In: Proceedings of the CSME Conference, Kingston, Canada (2002)
7. Kerle, H., Pittschellis, R., Corves, B.: Einführung in die Getriebelehre: Analyse und Synthese ungleichmäßig übersetzender Getriebe; mit 23 Tafeln sowie 29 Aufgaben mit Lösungen. Teubner, Wiesbaden, 3., bearb. und erg. aufl. edition (2007) ISBN 978-3-835-10070-1
8. Volmer, J.: Getriebetechnik, 2nd edn. Verlag Technik, Berlin (1995)

9. Hesselbach, J., Frindt, M.: Structural classification and systematic design of machines basing on parallel structures. In: 31st International Symposium on Robotics, pp. 65–70 (2000)
10. Frindt, M.: Modulbasierte Synthese von Parallelstrukturen für Maschinen in der Produktionstechnik. In: Schriftenreihe des Instituts für Werkzeugmaschinen und Fertigungstechnik der TU Braunschweig. Vulkan-Verlag, Essen (2001) ISBN 978-3802786594
11. Hunt, K.H.: Structural kinematics of in-parallel-actuated structure-arms. Journal of Mechanisms, Transmissions, and Automation in Design 105(4), 705–712 (1983)
12. Pierrot, F., Dauchez, P., Fournier, A.: Towards a fully-parallel 6 DOF robot for high speed applications. In: IEEE International Conference on Robotics and Automation, pp. 1288–1293 (1991)
13. Gough, V.E., Whitehall, S.G.: Universal tire test machine. In: 9th International Technical Congress F.I.S.I.T.A., pp. 117–135 (1962)
14. Griffis, M., Crane, C., Duffy, J.: A smart kinestatic interactive platform. In: Advances in Structure Kinematics and Computational Geometry, pp. 459–464. Kluwer Academic Publishers, Dordrecht (1994)
15. Hebsacker, M.: Effektiver Fräsen mit sechs Beinen. In: Schweizer Präzisions-Fertigungstechnik, pp. 28–30. Carl Hanser Verlag (1997)
16. Merlet, J.-P., Gosselin, C.M.: Nouvelle architecture pour un manipulateur parallèle à 6 degrés de libertè. Mechanism and Machine Theory 26(1), 77–90 (1991)
17. Hunt, K.H.: Geometry of robotic devices. Mechanical Engineering Transactions 7(4), 213–220 (1982)
18. Frindt, M., Kerle, H., Plitea, N.: Penta - Vorstellung eines parallelen Maschinenkonzepts mit fünf Bewegungsfreiheiten. VDI Berichte 1427, 15–34 (1998)
19. Plitea, N., Hesselbach, J., Kerle, H.: Penta - a machine concept with five degrees of freedom - rules of design and peculiarities. In: Proceedings of the 9th DAAAM International Symposium, Cluj-Napoca, Romania, pp. 397–398 (1998)
20. Gosselin, C.M.: Determination of the workspace of 6-DOF parallel manipulators. Journal of Mechanical Design 112(3), 331–336 (1990)
21. Gosselin, C.M., Zhang, D.: Some implications for parallel kinematic machine design based on kinetostatic model. In: Proceedings of the ASME 2000 Design Engineering Technical Conference, Baltimore, Maryland, USA (2000)
22. Hesselbach, J., Plitea, N., Kerle, H., Frindt, M.: Bewegungsvorrichtung vom Freiheitsgrad 5 mit Parallelstruktur (2000)
23. Clavel, R.: Delta, a fast robot with parallel geometry. In: 18. International Symposium on Industrial 1998, Lausanne, Switzerland, pp. 91–100 (1988)
24. Demaurex, M.O.: Le robot delta dans l'industrie. In: Internationales Parallelkinematik-Kolloquium IPK, Zurich, Switzerland, pp. 55–63 (1998)
25. Lambert, M.: Polyarticulated retractile mechanism (1987)
26. Stewart, D.: A platform with six degrees of freedom. In: Proceedings of the Institution of Mechanical Engineers (1965)
27. Tahmasebi, F., Tsai, L.W.: Closed-form direct kinematics solution of a new parallel minimanipulator. Journal of Mechanical Design 116, 1141–1147 (1994)
28. Romiti, A., Sorli, M.: Flexible sensorized micro assembly by a small parallel manipulator. In: Proceedings of the International FAMOS Seminar, Besancon, France (1990)
29. Merlet, J.-P.: Efficient design of parallel robots. VDI Berichte 1427, 1–13 (1998)
30. Angeles, J.: The qualitative synthesis of parallel manipulators. In: Proceedings of the International Workshop on Fundamental Issues and Future, Quebec City, Canada, pp. 160–170 (2002)
31. Salisbury, J.K., Craig, J.J.: Articulated hands: Force control and kinematic issues. International Journal of Robotics Research 1(1), 4–17 (1982)

32. Kirchner, J.: Mehrkriterielle Optimierung von Parallelkinematiken, 1st edn. Berichte aus dem IWU, vol. 12. Verlag Wissenschaftliche Scripten, Zwickau (2001) ISBN 3928921673
33. Bowling, A., Khatib, O.: The dynamic capability equations: A new tool for analyzing robotic manipulator performance. IEEE Transactions on Robotics and Automation 21(1), 115–123 (2005)
34. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Fonseca, V.G.: Performance assessment of multiobjective optimizers. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)

Automated Generation of Efficient Real-Time Code for Inverse Dynamic Parallel Robot Models

Michael Rose

Abstract. In this paper, the principal underlying equations and a method for real-time code generation are discussed to obtain fast inverse dynamic models of robot structures for the so-called computed torque control. This sophisticated trajectory control algorithm linearizes the highly non-linear dynamic properties of the underlying plant to increase the path tracking precision. To fit the calculations into the general small time slot of real-time control algorithms, special care is needed in the code generation phase. The application to the TRIGLIDE robot structure is finally presented.

1 Introduction

Parallel robot structures are special multi-body systems with closed kinematic loops, several acutator drives, and typically one end-effector attached to the manipulator base platform to fulfill the specific tasks, e.g., handling and assembly. To simulate such mechanical body scenarios, kinematic and dynamic models are indispensable; but even for real-time control tasks, a so-called inverse dynamic model is of great advantage enabling either the reduction of contouring errors during trajectory traversal or higher accelerations with shorter cycle times under consideration of a given trajectory precision.

In this article, a method for the automatic generation of a real-time capable inverse dynamic model of parallel structures is described. The only prerequisite will be that the kinematics of the underlying closed loop mechanical system can be solved either analytically or numerically. Force models of this kind are needed by the so-called *computed torque method*, which is described in Sect. 3. Two examples of parallel structures will be discussed more thoroughly; they are displayed

Michael Rose
German Aerospace Center (DLR), Institute of Composite Structures and Adaptive Systems, Lilienthalplatz 7, 38108 Braunschweig, Germany
e-mail: michael.rose@dlr.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 39–57.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

Fig. 1. Example parallel structures: **a** HEXA manipulator. **b** TRIGLIDE manipulator.

in Fig. 1. These parallel robots are demonstrators of the *"Collaborative Research Centre SFB 562"*.

For the theoretical investigation, variational principles are used. Using the kinematic solution, a tree connectivity of the single bodies is traversed to assemble the mass matrices and force vectors of the dynamic model. To be real-time capable, this is done by a symbolic code generation tool, developed especially for that purpose to yield highly optimized C-code.

2 Related Work

There are several highly sophisticated tools available to solve arbitrary multi-body systems with or without kinematic loops. Some of them are also capable to export program code of the underlying equations [1, 2]. Indeed, such tools are used here to validate the generated program code by comparing the simulation output of the newly developed tools with the corresponding output of an equivalent model in the SIMPACK©-multi-body software. The development of independent tools is justified, because this allows to solve some sort of problems more effectively. Especially the calculation of inverse dynamics can be implemented as fast as possible if direct control of the kind of equations being exported is granted. Therefore, it is advantageous to create these equations with specialized and adaptable tools.

There are several good textbooks that explain the underlying equations of multi-body systems in great detail. In [3, 4], the single rigid/elastic body equations as well as the joints and constraints of a complete system are derived. In [5], the numerical aspects of computations for such systems are in focus, especially by using backward differentiation formulas (BDF) for the underlying differential algebraic equations (DAE). A short summary on the equations for elastic multi-body systems is given in [6, 7].

The example robot structures (e.g. the HEXA-structure, the TRIGLIDE-structure, and also the simple two dimensional FIVE-BAR test structure) are discussed in [8, 9, 10, 11, 12, 13, 14] with special focus on the smart actuator technology and the control of structural vibrations. The full details of the kinematic and dynamic equations for the HEXA [15] and for the TRIGLIDE [16] structures are described in technical

reports. A focus on the computed torque approach for the HEXA parallel structure can be found in [17].

It is known, that the direct kinematic problem of parallel robot systems like the HEXA-structure or the Stewart-Gough platform has no closed form solution. But there are theoretical investigations based on algebraic methods and the so-called Grassmann geometry for the latter robot structure. In [18] the maximum number of possible configurations of the Stewart-Gough platform is shown to be fourty by a combination of algebraical and numerical methods. An algorithm based on the calculation of the roots of a 40th degree polynomial to solve the direct kinematic problem of this platform is presented in [19]. A general discussion of singular configurations of parallel manipulators based on the Grassmann geometry can be found in [20]. Finally there are well-known robust numerical tools available to solve the kinematic equations along a trajectory by iterative tracking procedures, see [21].

3 The Computed Torque Method

The motivation to implement fast inverse dynamic models of parallel structures is based on the fact that control algorithms often benefit from using the aforementioned force based plant models. Either the achievable accelerations can be increased, leading to shortened cycle times, or alternatively, the tracking error of the manipulator's trajectory can be decreased. One favorable approach is the so-called *computed torque method*, which will be shortly described in this section.

3.1 Plant Model of the Robot Structure

The approach is based on the general dynamic eq. (1)

$$\boldsymbol{M}_{(\boldsymbol{q})}\ddot{\boldsymbol{q}} + \boldsymbol{\xi}_{(\boldsymbol{q},\dot{\boldsymbol{q}})} = \boldsymbol{\tau} - \boldsymbol{G}_{(\boldsymbol{q})}{}^{-T}\boldsymbol{f}, \qquad \boldsymbol{\xi}_{(\boldsymbol{q},\dot{\boldsymbol{q}})} = \boldsymbol{C}_{(\boldsymbol{q},\dot{\boldsymbol{q}})}\dot{\boldsymbol{q}} \tag{1}$$

with the free joint variables $\boldsymbol{q}$ of the drives, the in general position-dependent mass matrix $\boldsymbol{M}$, the drive torques or forces $\boldsymbol{\tau}$ choosen by the drive control algorithm, external forces $\boldsymbol{f}$ at the manipulator's end-effector and the combined centrifugal, gyroscopic, and friction-based forces $\boldsymbol{\xi}$ depending on the position and velocity state. Usually, the external forces are unknown, unless they are estimated with additional sensors or with an observer-based approach, but contact forces and the moved masses and therefore the inertial forces are generally the main contributions to the reaction of the structure with respect to drive actuation. There are two main calculation modes possible with these equations:

- *Inverse dynamics mode*: Based on the trajectory data with positions $\boldsymbol{q}$, velocities $\dot{\boldsymbol{q}}$, and accelerations $\ddot{\boldsymbol{q}}$, the drive torques and forces $\boldsymbol{\tau}$ are calculated. This is the relevant mode for the actual discussion.
- *Direct dynamics mode*: Based on all forces, the trajectory data of the resulting movement on position, velocity, and acceleration level is calculated. This is the foundation of any simulation in the time domain.

To be able to calculate all values of these models, different kinematic procedures are needed. The two main evaluation strategies are given by

- *Inverse kinematic problem (IKP)*: The drive displacements $\boldsymbol{q}$ are calculated from the trajectory of the manipulator $\boldsymbol{p}$. It is the basic kinematic calculation for drive-based control with included dynamic effects; it often is analytically solvable for parallel structures with many chains. For serial structures with many serial chains, this procedure generally has to be solved numerically. The derivated equation $\dot{\boldsymbol{q}} = \boldsymbol{G}\dot{\boldsymbol{p}}$ defines the jacobian $\boldsymbol{G}_{(\boldsymbol{q})}$, which is also used in eq. (1).
- *Direct kinematic problem (DKP)*: The manipulator position $\boldsymbol{p}$ is calculated from the drive positions $\boldsymbol{q}$. The derivated equation defines the corresponding jacobian, which is given by $\boldsymbol{G}^{-1}_{(\boldsymbol{q})}$ if kinematic singularities are neglected. This is normally the challenging part of kinematic calculations for parallel structures in contrast to serial structures with actuator drives in all chain parts. It is needed for control algorithms, that perform force control at the manipulator or for simulations, which are based on the drive states $\boldsymbol{q}$.

Note that due to the energy considerations $\boldsymbol{\tau}^T\boldsymbol{q} = \boldsymbol{f}^T\boldsymbol{p}$, a transformation of the forces $\boldsymbol{f} = \boldsymbol{G}^T\boldsymbol{\tau}$ is always possible and already used in equation (1). Other kinematic calculations are also needed in practical implementations, because the positions and velocities of intermediate bodies have to be known to incorporate their dynamic effect on the whole system.

3.2 Linearization Using Inverse Dynamic Models

Figure 2 shows the principal approach of the computed torque method with the HEXA-robot as plant structure. It is basically a linearization of the non-linear model equations. The trajectory data $\boldsymbol{p}_0$, $\dot{\boldsymbol{p}}_0$, and $\ddot{\boldsymbol{p}}_0$ from the trajectory planning module is given as input at the current time t. With the *IKP*-module, these values are transformed to the drive level. By using the inverse dynamic model, the neccessary torques to be fed to the robot drives can be estimated with good accuracy and are supplied directly to the drives. In an exact world, this would be sufficient to let the robot structure follow the desired trajectory. But due to unknown external forces, inexact parameters, and other unavoidable inaccuracies at the modelling level, a control algorithm, denoted as *PD* in the diagram, is used to alleviate these effects.

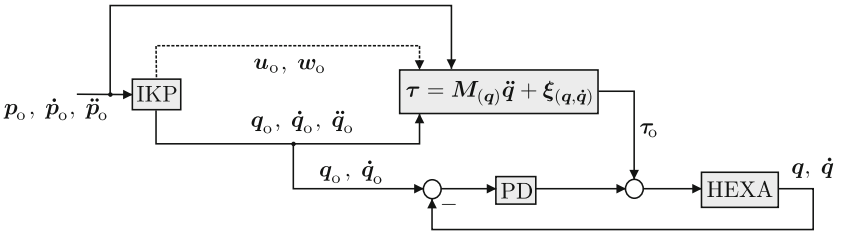


Fig. 2. Dynamic force compensation with the computed torque method.

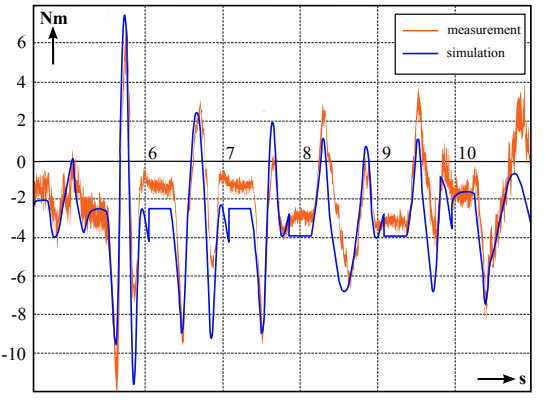


Fig. 3. Example of the accuracy of an inverse dynamic model based on the HEXA-robot structure. A plot of one drive torque (in Nm) over time (in seconds) is shown.

If the actual deviation from the pre-planned trajectory is not too high, the structure seen by the controller is quasi linearized due to the compensation with the inverse dynamic model. An example plot of the deviations of the calculated torques at one drive of the real HEXA-structure with respect to measurement is given in Fig. 3. There is obviously some error mainly due to neglecting friction, but the overall correlation is sufficient to achieve accelerations of up to $10g$ acceleration.

4 Principal Robot Equations

The strategy of deriving the equations of motion from mechanical systems is given by the so-called multi-body formalism. There are several good textbooks on this topic as explained in Sect. 2. This section will focus on that part of the theory, which is sufficient to build an inverse dynamic model. It is divided into the kinematic and the dynamic aspect of the equations.

4.1 Kinematic Modelling

One big drawback in automatic model creation is given by the fact that many kinematic systems can be solved only numerically. Numeric solutions may suffer from convergence problems, and it is very hard to fit them in the control cycle slot Δt of real-time systems, because the iteration to converge usually needs program loops with a priori unknown number of cycles. The approach taken here is twofold. For simulation purposes, a general method is described to assemble the kinematic procedure, which has to be solved numerically. For many real systems, the *IKP* can be solved analytically. Because this module is the one needed for the inverse dynamics, it can be used instead of the general kinematic module and has to be supplied manually. Fortunately, the tedious derivation of the kinematic relations at the velocity and acceleration levels can completely be done symbolically, because the system is always linear with respect to velocities. Manual differentiation of these equations is error prone. An acceptable approach in this case is the use of automatically created

kinematic equations at the velocity and acceleration levels using position values, which are delivered by the module the user has provided manually. If an analytic solution exists, the kinematic relations at the position level are often quite handy. This is the case for all the investigated structures of the SFB 562.

To describe all parts of a mechanical system appropriately, different coordinate systems have to be introduced. The number of unknown parameters can be significantly reduced if relative coordinates between body fixed reference systems are used. The mechanical structure of the robot has to be described as tree structure. Each single rigid component is a node in this tree, and the edges are given by the joints between the components. The closed loops have to be treated separately leading to additional constraint forces with zero virtual energy. In order to describe the kinematic relationship between the bodies due to the joints, some coordinate transformations $\mathcal{O}_Q = \mathcal{B} \cdot \mathcal{O}_P$ will be used. Coordinate system $\mathcal{O}_Q$ can be calculated by applying the transformation $\mathcal{B}$ to coordinate system $\mathcal{O}_P$. Only orthogonal transformations are needed, and coordinate systems are given by an origin and three axes. Transformations can be conveniently represented by using homogeneous coordinates. Of special interest are the three elementary translations

$$\mathcal{T}_x[a], \quad \mathcal{T}_y[b], \quad \mathcal{T}_z[c]$$

in axial directions x, y, or z by the values a, b, or c respectively and the three elementary rotations

$$\mathcal{R}_x[a], \quad \mathcal{R}_y[b], \quad \mathcal{R}_z[c]$$

around the coordinate axes by angles a, b, or c given in radians. The special inertial system $\mathcal{O}$ is always the root of the constructed kinematic tree. If the kinematic problem is solved at the position level, these transformations can be automatically calculated to derive the complicated equations at the velocity or acceleration levels in combination with techniques known as automated differentiation. We will see concrete examples on how to build such a tree in Sect. 5.

The use of elementary translations and rotations is of course only one possibility to construct the kinematic tree. It is the most simple approach, and the model design process is quite comfortable in this framework. Other approaches like the *Denavit-Hartenberg-Transformation* or an extension of this notation formalism, as discussed in [22], are especially designed to reduce the number of joint parameters for the robot mechanism. For pure kinematic solvers, this reduction can be quite advantageous, but here the gain in efficiency is usually not as high, because the inertia terms of each body have to be calculated anyway. This requires the evaluation of many additional terms in the kinematic chains, which could otherwise be skipped. In contrast, explicit solutions of solvable kinematic problems are created by careful analysis of the kinematic problem with individual solution strategies, based on a global view of the problem and not on a tree structure.

4.2 Dynamic Modelling

An application of the classical Jourdain principle [4] for mechanical structures balances the internal virtual power

$$\sum_k \delta P_k = \sum_i \delta \dot{q}_i \, \tau_i \tag{2}$$

of individual bodies and their joints with the supplied external virtual power due to drive forces or manipulator actions. Additional terms, like friction terms, are neglected in this considerations for simplicity, but they can easily be incorporated if the type of friction is known. Here, the drive angles q_i and the drive torques τ_i in case of the HEXA-structure or the transversal positions and forces in case of the TRIGLIDE-structure are the only external units. If elasticities are neglected and no energy based joints are used, P_k represents the virtual power of the inertia forces at each body of the system. This is given for rigid bodies with mass m, center of gravity $\boldsymbol{c}$, and inertia tensor $\boldsymbol{J}$ by equation (3).

$$\delta P = \begin{bmatrix} \boldsymbol{\delta v} \\ \boldsymbol{\delta \omega} \end{bmatrix}^T \left[\begin{bmatrix} m\boldsymbol{I} & m\tilde{\boldsymbol{c}}^T \\ m\tilde{\boldsymbol{c}} & \boldsymbol{J} \end{bmatrix} \begin{bmatrix} \boldsymbol{a} - \boldsymbol{g} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} + \begin{bmatrix} m\tilde{\boldsymbol{\omega}}\tilde{\boldsymbol{\omega}}\boldsymbol{c} \\ \tilde{\boldsymbol{\omega}}\boldsymbol{J}\boldsymbol{\omega} \end{bmatrix} \right]. \tag{3}$$

All occuring vectors and tensors can be related to an arbitrary but unique reference system. The most appropriate system here is a body fixed reference system with absolute velocity $\boldsymbol{v}$, absolute acceleration $\boldsymbol{a}$, and absolute angular velocity $\boldsymbol{\omega}$. The tilde operator maps a vector to a 3×3-matrix to evaluate the outer vector product as a matrix vector multiplication, as in $\tilde{\boldsymbol{\omega}}\boldsymbol{c} = \boldsymbol{\omega} \times \boldsymbol{c}$. All internal energy in a rigid body system without springs and damping results from inertia terms and can be calculated by summing up the inertia energies of eq. (3) for each single body if the motion of each body is fully described in terms of $\boldsymbol{a}$ and $\boldsymbol{\omega}$, which are automatically calculated by the kinematic tree of the robot system. The unknown external forces or torques of the drives can subsequently be calculated by eq. (2).

5 Example Structures

To concretize the described approach of obtaining an inverse dynamic model of parallel robot structures, two examples are investigated in the following, namely the HEXA-structure and the TRIGLIDE-structure. Both structures as displayed in Fig. 1 are extensively described in [15, 16, 14]. Therefore, the following discussion is just a brief review of the main ideas to emphasize the overall procedure. In [11], the inclusion of elastic components into such robot systems is discussed. For general ideas about multi-body systems with elasticities, see also [3, 4, 6, 7, 23].

5.1 The HEXA-Robot

This structure is a six degree of freedom parallel structure. Therefore, the end-effector can be positioned in space by three translational and three rotational

independent variables. On the other side, six torque generating direct drive motors are mounted rigidly on the base platform to control all six degrees of freedom. In contrast to the *DKP*, the *IKP* is solvable by some simple considerations with spherical angles for each of the six kinematical chains. In this example, the manually solved *IKP* will be explained.

The geometric parameters of this structure are collected in Table 1. The acceleration $g = 9.81\text{m/s}^2$ acts in negative z-direction of the world system $\mathcal{O}$. The coordinate system $\mathcal{O}_i$ of the i-th rotational drive is positioned by translation with h in z-direction and rotation with γ_i around the z-axis on the drive platform for $i \in \{1, \ldots, 6\}$. The symmetric configuration is given by $\gamma_1 = \gamma_6 = 0$, $\gamma_2 = \gamma_3 = \frac{2\pi}{3}$, and $\gamma_4 = \gamma_5 = \frac{4\pi}{3}$. The home position of the manipulator platform is parallel to the world system, translated by h_P in z-direction. The i-th drive operates at point A_i with coordinates relative to the i-th drive coordinate system given by

$$[A_i]_i = [r_{B_i}, b_{B_i}, 0]^T \quad \text{with} \quad b_{B_i} = (-1)^{i+1} b_B.$$

Correspondingly, the i-th global joint of the manipulator platform at point C_i has the relative coordinates

$$[C_i]_{P_i} = [r_{P_i}, b_{P_i}, 0]^T \quad \text{with} \quad b_{P_i} = (-1)^{i+1} b_P,$$

whereas the coordinate system $\mathcal{O}_{P_i}$ results from a rotation about the angle γ_i around the z-axis of the reference system from the manipulator platform $\mathcal{O}_P$. Point T denotes the position of the end-effector with constant coordinates $[T]_P$ with respect to $\mathcal{O}_P$. The corresponding system $\mathcal{O}_T$ with origin T is always parallel to $\mathcal{O}_P$. The crank with length r_i will rotate due to the i-th drive by the drive angle q_i at point A_i around the y-axis of $\mathcal{O}_i$. Therefore, the point B_i of the i-th cardan joint is given by

$$[B_i]_i = \begin{bmatrix} x_{B_i} \\ y_{B_i} \\ z_{B_i} \end{bmatrix} = \begin{bmatrix} r_{B_i} \\ b_{B_i} \\ 0 \end{bmatrix} + r_i \begin{bmatrix} \cos q_i \\ 0 \\ -\sin q_i \end{bmatrix}. \tag{4}$$

The rod with length d_i has the end points B_i and C_i. Introducing the cardan angles u_i and w_i of rotation around the y-axis and z-axis respectively, the direction vector of the i-th rod can be calculated as

$$[C_i - B_i]_i = d_i \begin{bmatrix} \cos(q_i + u_i)\cos(w_i) \\ \sin(w_i) \\ -\sin(q_i + u_i)\cos(w_i) \end{bmatrix}. \tag{5}$$

The six end-effector states $\boldsymbol{p} = [x_e, y_e, z_e, \psi_e, \theta_e, \varphi_e]^T$ describe the position and orientation of the movable platform with

$$[T]_O = \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \quad \text{and} \quad \underbrace{\text{Rot}_{O,P}}_{=\boldsymbol{R}} = \begin{bmatrix} c\varphi\, c\theta & c\varphi\, s\theta\, s\psi - s\varphi\, c\psi & c\varphi\, s\theta\, c\psi + s\varphi\, s\psi \\ s\varphi\, c\theta & s\varphi\, s\theta\, s\psi + c\varphi\, c\psi & s\varphi\, s\theta\, c\psi - c\varphi\, s\psi \\ -s\theta & c\theta\, s\psi & c\theta\, c\psi \end{bmatrix}, \tag{6}$$

Table 1. Geometric parameters of the HEXA-structure.

Description	Symbol	Value
Distance drive platform to world system	h	0 mm
Home position of manipulator platform	h_P	-500 mm
Distance drive crank to center of base platform	r_{B_i}	360 mm
Distance of global joint to center of manipulator platform	r_{P_i}	51,96 mm
Pairwise distance of drives and global joints	$2b$	103,2 mm
Crank length	r_i	240 mm
Rod length	d_i	564 mm
x-coordinate of end-effector	x_T	0 mm
y-coordinate of end-effector	y_T	0 mm
z-coordinate of end-effector	z_T	-120 mm

the global rotation matrix to transform the world system into the local system $\mathcal{O}_P$. Here, some abbreviations of the form $c\varphi = \cos(\varphi_e)$, $s\psi = \sin(\psi_e)$, etc. are used. $\boldsymbol{R}$ is given by the elementary rotations

$$\boldsymbol{R} = \boldsymbol{R_3}\,\boldsymbol{R_2}\,\boldsymbol{R_1} = \mathcal{R}_z[\varphi_e]\mathcal{R}_y[\theta_e]\mathcal{R}_x[\psi_e],$$
$$\boldsymbol{R_1}^T\dot{\boldsymbol{R}}_{\boldsymbol{1}} = \dot{\psi}\,\tilde{\boldsymbol{e}}_1, \quad \boldsymbol{R_2}^T\dot{\boldsymbol{R}}_{\boldsymbol{2}} = \dot{\theta}\,\tilde{\boldsymbol{e}}_2, \quad \boldsymbol{R_3}^T\dot{\boldsymbol{R}}_{\boldsymbol{3}} = \dot{\varphi}\,\tilde{\boldsymbol{e}}_3.$$

The vector $\boldsymbol{e}_i$ denotes the i-th unit vector. The three angles describe the yaw around the z-axis by φ_e, the pitch around the y-axis by θ_e, and the rolling around the x-axis by ψ_e of the end-effector with respect to the world system $\mathcal{O}$. For the global joint points C_i, the coordinate representations

$$[C_i]_P = \mathcal{R}_z[\gamma_i]\,[C_i]_{P_i}\,,$$
$$[C_i - P]_O = \mathrm{Rot}_{O,P}\,[C_i]_P\,,$$
$$[C_i]_O = \mathcal{R}_z[\gamma_i]\,[C_i]_i + [0,0,h]^T$$

are obtained. The relation

$$[\boldsymbol{T} - \boldsymbol{P}]_O = \mathrm{Rot}_{O,P}\,[\boldsymbol{T}]_P$$

supplies the coordinates of $[P]_O$. With

$$[T]_P = \begin{bmatrix} x_T \\ y_T \\ z_T \end{bmatrix}, \quad \boldsymbol{D}_i = \mathcal{R}_z[\gamma_i] = \begin{bmatrix} \cos\gamma_i & -\sin\gamma_i & 0 \\ \sin\gamma_i & \cos\gamma_i & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{c}_i = \boldsymbol{D}_i \begin{bmatrix} r_{P_i} \\ b_{P_i} \\ 0 \end{bmatrix}, \quad \boldsymbol{v}_i = \boldsymbol{c}_i - \begin{bmatrix} x_T \\ y_T \\ z_T \end{bmatrix},$$

we can derive

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = [C_i]_i - \begin{bmatrix} r_{B_i} \\ b_{B_i} \\ 0 \end{bmatrix} = \boldsymbol{D}_i{}^T \left[\begin{bmatrix} x_e \\ y_e \\ z_e - h \end{bmatrix} + \boldsymbol{R}\, \boldsymbol{v}_i \right] - \begin{bmatrix} r_{B_i} \\ b_{B_i} \\ 0 \end{bmatrix}. \tag{7}$$

This is the vector connecting the points A_i and C_i with respect to $\mathcal{O}_i$. The essential part of the *IKP* can now be explained very easily. For the dynamic model, the moment of inertia of the rods in rod direction are neglected. Therefore, only the drive angles $\boldsymbol{q}$ have to be calculated from the independent states $\boldsymbol{p}$. The cardan angles u_i and w_i are not needed. Taking squares, summing the coordinates from equation (5), substitution of B_i from equation (4), and using $[C_i - A_i]_i = [x_i, y_i, z_i]^T$ delivers the kinematic relationship

$$x_i \cos q_i - z_i \sin q_i = \alpha_i, \tag{8}$$

with the abbreviation

$$\alpha_i = \frac{1}{2r_i} \left[x_i^2 + y_i^2 + z_i^2 + r_i^2 - d_i^2 \right].$$

Figure 4 shows the projection of the crank and rod along the y-axis onto the xz-plane of system $\mathcal{O}_i$ (the index i has been neglected for simplicity). The drive angle q_i has to be taken with an opposite sign, because the y-axis points away from the observer. The angle η_i is now given by

$$\eta_i = \arctan(z_i/x_i) \quad \text{or} \quad \beta_i \cos\eta_i = x_i,\ \beta_i \sin\eta_i = z_i \quad \text{with} \quad \beta_i = \sqrt{x_i^2 + z_i^2}.$$

If x_i and z_i in eq. (8) are replaced by polar coordinates, the relation

$$\beta_i \cos(q_i + \eta_i) = \alpha_i$$

follows. Thus, q_i is determined by

$$q_i = \pm \arccos \frac{\alpha_i}{\beta_i} - \eta_i = \arctan\left(\pm \sqrt{\beta_i^2 - \alpha_i^2} \,\Big/\, \alpha_i \right) - \eta_i.$$

The negative sign corresponds to a configuration as shown in Fig. 4, if an outer configuration of the crank/rod leg is choosen. This is an advantageous configuration to avoid undesired collisions of different drive chains. Configurations can only be

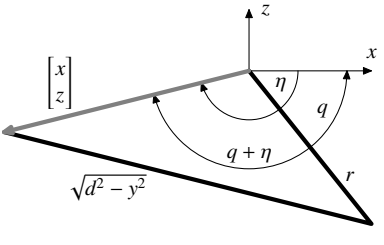


Fig. 4. Derivation of q by projecting crank and rod in the xz-plane.

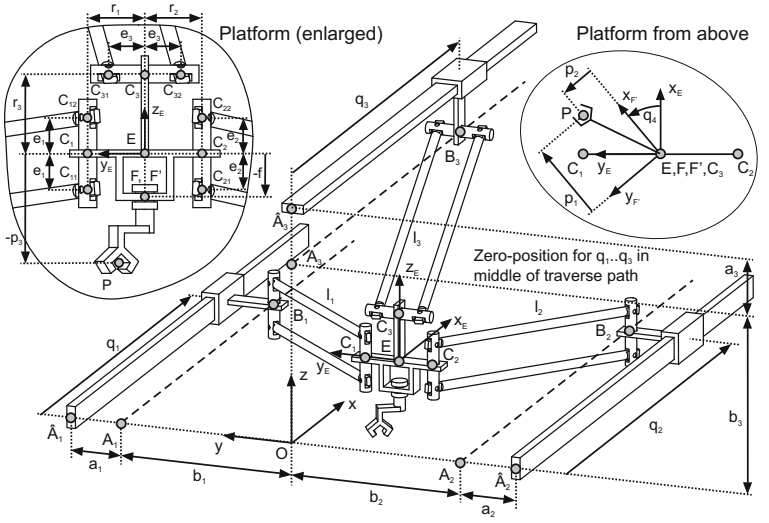


Fig. 5. Schematic view of the TRIGLIDE-structure.

changed by crossing a singularity of the parallel structure. This is precluded here for the derived dynamic model by choosing a negative sign for all six drive chains.

5.2 The TRIGLIDE*-Robot*

The TRIGLIDE-structure is a three degree of freedom parallel structure in combination with an additional serial rotational degree of freedom at the end-effector platform. The workspace consists of two distinct configuration spaces. Both of them are usable by passing through a second order singularity (see [9]). It has the advantage that the *IKP* as well as the *DKP* can be solved analytically. This allows the free choice of independent variables as drive positions or manipulator positions. A schematic view with an explanation of the kinematic parameters is given in Fig. 5. In this example, the focus lies in the construction of the kinematic tree. The geometric properties of the real structure are given in Table 2. The special $\mathcal{O}$-system is the inertial base system. The initial system with $q_i = 0$ of each linear drive is given by

$$\begin{aligned}
\mathcal{O}_1 &= \mathcal{T}_y[-b_1 - a_1] \cdot \mathcal{R}_x[180^\circ] \cdot \mathcal{O}, \\
\mathcal{O}_2 &= \mathcal{T}_y[-b_2 - a_2] \cdot \mathcal{O}, \\
\mathcal{O}_3 &= \mathcal{T}_y[-b_3 - a_3] \cdot \mathcal{R}_x[-90^\circ] \cdot \mathcal{O}.
\end{aligned}$$

A translational shift by the drive coordinates q_i leads to the reference system of a drive slider

$$\mathcal{O}_{\mathrm{AS}_i} = \mathcal{T}_x[q_i] \cdot \mathcal{O}_i.$$

Table 2. Geometric parameters of the TRIGLIDE-structure

Description	Symbol	Value
Distance of the rotational joint to the drive axes	a_i	105 mm
Distance of the drive axes to the x-axis	b_i	$\{379, 379, 415\}$ mm
Rod length	ℓ_i	600 mm
Distance of the rotational platform joints to point E	r_i	$\{79, 79, 110\}$ mm
Half length of joint axes	e_i	50 mm
Coordinates of point P with respect to point E	$[p_1, p_2, p_3]^T$	0 mm

With respect to these systems, the reaction forces and moments on the drive sliders are calculated. The rotational joints GS_i on the drive sides are connected by revolute joints

$$\begin{aligned}\mathcal{O}_{\mathrm{AS(GS)}_1} &= \mathcal{R}_y[180^\circ] \cdot \mathcal{T}_y[a_1] \cdot \mathcal{O}_{\mathrm{AS}_1},\\ \mathcal{O}_{\mathrm{AS(GS)}_2} &= \mathcal{T}_y[a_2] \cdot \mathcal{O}_{\mathrm{AS}_2},\\ \mathcal{O}_{\mathrm{AS(GS)}_3} &= \mathcal{T}_y[a_3] \cdot \mathcal{O}_{\mathrm{AS}_3}\end{aligned}$$

to the drive sliders. The dependent internal angles γ_i now define the reference systems

$$\mathcal{O}_{\mathrm{GS}_i} = \mathcal{R}_z[\gamma_i] \cdot \mathcal{O}_{\mathrm{AS(GS)}_i}$$

of the cylindric joint axes. The upper rods SP_i and the lower rods SN_i are also connected by revolute joints as given by

$$\mathcal{O}_{\mathrm{GS(SP)}_i} = \mathcal{R}_x[-90^\circ] \cdot \mathcal{T}_z[e_i] \cdot \mathcal{O}_{\mathrm{GS}_i}, \quad \mathcal{O}_{\mathrm{GS(SN)}_i} = \mathcal{R}_x[-90^\circ] \cdot \mathcal{T}_z[-e_i] \cdot \mathcal{O}_{\mathrm{GS}_i}.$$

Their reference systems can be determined by a rotation with the passive joint angles β_i from the solution of the kinematic problem by the transformations

$$\mathcal{O}_{\mathrm{SP}_i} = \mathcal{R}_x[\beta_i] \cdot \mathcal{O}_{\mathrm{GS(SP)}_i}, \quad \mathcal{O}_{\mathrm{SN}_i} = \mathcal{R}_x[\beta_i] \cdot \mathcal{O}_{\mathrm{GS(SN)}_i}.$$

All the reaction forces of the rods are given with respect to these reference systems. Another set of transformations

$$\mathcal{O}_{\mathrm{SP(GP)}_i} = \mathcal{R}_y[180^\circ] \cdot \mathcal{T}_z[\ell_i] \cdot \mathcal{O}_{\mathrm{SP}_i}, \quad \mathcal{O}_{\mathrm{SN(GP)}_i} = \mathcal{R}_y[180^\circ] \cdot \mathcal{T}_z[\ell_i] \cdot \mathcal{O}_{\mathrm{SN}_i}$$

connects the rods to the systems

$$\mathcal{O}_{\mathrm{GP(SP)}_i} = \mathcal{R}_x[-90^\circ] \cdot \mathcal{T}_z[e_i] \cdot \mathcal{O}_{\mathrm{GP}_i}, \quad \mathcal{O}_{\mathrm{GP(SN)}_i} = \mathcal{R}_x[-90^\circ] \cdot \mathcal{T}_z[-e_i] \cdot \mathcal{O}_{\mathrm{GP}_i}$$

of the hinge shafts with respect to the manipulator platform. Because of the parallelogram structure, the same angles β_i are used here to derive the connections

$$\mathcal{O}_{\mathrm{GP(SP)}_i} = \mathcal{R}_x[\beta_i] \cdot \mathcal{O}_{\mathrm{SP(GP)}_i}, \quad \mathcal{O}_{\mathrm{GP(SN)}_i} = \mathcal{R}_x[\beta_i] \cdot \mathcal{O}_{\mathrm{SN(GP)}_i}.$$

Because of the overdetermined structure, reaction forces can only be determined in these parallelograms if additional elasticities are build into the dynamic model. For an inverse dynamic model however, this is not neccessary. Using the joint angles γ_i again, the transformations

$$\mathcal{O}_{\mathrm{GP}_i} = \mathcal{R}_z[\gamma_i]\mathcal{O}_{\mathrm{AP(GP)}_i}$$

lead to the systems $\mathcal{O}_{\mathrm{AP(GP)}_i}$, which are rigidly connected to the end-effector platform. Again, we use a result from the kinematic analysis about the identity $\gamma_{\mathrm{GP}_i} = \gamma_{\mathrm{GS}_i}$. The reference system of the end-effector platform $\mathcal{O}_{\mathrm{AP}}$ is now obtained by

$$\begin{aligned}
\mathcal{O}_{\mathrm{AP(GP)}_1} &= \mathcal{T}_y[r_1] \cdot \mathcal{O}_{\mathrm{AP}}, \\
\mathcal{O}_{\mathrm{AP(GP)}_2} &= \mathcal{T}_y[r_2] \cdot \mathcal{R}_z[180°] \cdot \mathcal{O}_{\mathrm{AP}}, \\
\mathcal{O}_{\mathrm{AP(GP)}_3} &= \mathcal{R}_y[180°] \cdot \mathcal{T}_y[r_3] \cdot \mathcal{R}_x[90°] \cdot \mathcal{O}_{\mathrm{AP}}.
\end{aligned}$$

Two of these relations are closed kinematic loops, which are only needed if the automatic solver is used for the kinematic problem. Another result from the manually performed kinematic analysis is the fact that the end-effector platform remains always parallel to its original orientation. In the coordinate system

$$\mathcal{O}_{\mathrm{F}} = \mathcal{T}_z[f] \cdot \mathcal{O}_{\mathrm{AP}}$$

all reaction forces and moments of the gripper are formulated. The gripper is connected to the platform by an additional serial actively driven revolute joint. The vertical position of F is only needed to calculate the mentioned reaction forces and moments. Therefore, the relative position of the end-effector or gripper is given by the transformation

$$\mathcal{O}_{\mathrm{EF}} = \mathcal{R}_z[\psi] \cdot \mathcal{O}_{\mathrm{AP}}.$$

Finally, the robot trajectories are describing the point P connected to the end-effector by the translation

$$\mathcal{O}_{\mathrm{P}} = \mathcal{T}_z[p_3] \cdot \mathcal{T}_y[p_2] \cdot \mathcal{T}_x[p_1] \cdot \mathcal{O}_{\mathrm{EF}}.$$

Some simplifications resulting from the manually solved *IKP*, which is not presented here, are embedded in these transformations. If a numerical solution of the kinematic problem is sufficient and no real-time code is needed, these simplifications are not neccessary and the equations can be fed directly into a software system as described in the next section, in order to obtain all dependent parameters from numerical iteration with some non-linear equation solver. There is some redundancy in the derived equations, but this is not a principle problem for these equations to be numerically solvable.

6 Generation of Efficient Real-Time Code

In order to develop software for the automatic processing of symbolic equations, a hierarchical structure of the software is very convenient and helps to modularize the individual tasks. This also greatly reduces the probability of errors and simplifies the creation of test scenarios for each constructed software module. Some intermediate results of the module test are displayed for the HEXA-robot and the TRIGLIDE-robot in Fig. 6.

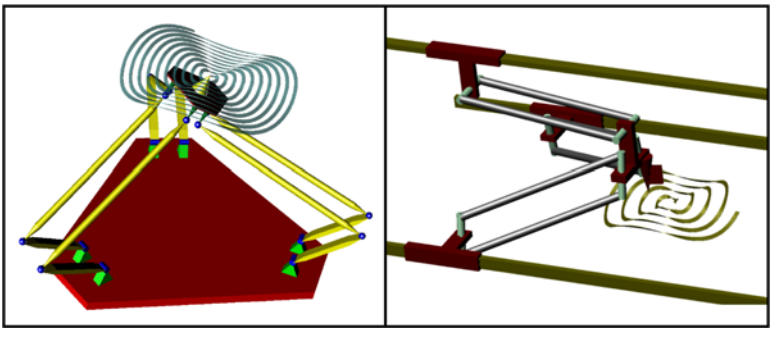


Fig. 6. Module test with different trajectories for the HEXA-robot and the TRIGLIDE-robot.

6.1 Hierarchical Software Structure

To create software toolboxes, which are easy in terms of maintainance, a hierarchical software structure is proposed. The chronological course of action is given by the following topics:

- Description of the kinematic and dynamic parallel structure in an application specific language.
- Translation of the application specific language in a basic tensor language, which describes the kinematic chains and the dynamic properties by means of a scripting language like PYTHON or PERL.
- Execution of the tensor program in the scripting interpreter to create a file with low-level terms, building the symbolic expressions.
- Translation of the low-level terms to a low-level C-program with appropriate interface code for the target system.
- Compilation in the framework of the real-time target system.
- Alternatively compilation into a dynamic link library, to be used in the simulation environment (in our case based on the matrix orientated software system MATLAB©).
- Use of the created module in application specific tasks, e.g., as a kinematic solver or as a force and torque calculator for specific trajectories.

6.2 Low-Level Term Processing

The low-level task in this process chain is given by a highly optimized C++-program with name *SymbolicCode*. In Fig. 7, the execution steps of the program *SymbolicCode* are displayed as a flow diagram. First, a term list is read from an external file, which was created by the tensor library of the upper software layer. Each line of this file consists of a single term. The syntax is very simple. The first element is the term number, starting from one. After the name of the term, separated by a tabulator char, the space separated arguments are listed. Table 3 describes some of the available terms. There are constants, parameters, and operators like summation (*Add*), trigonometric operators (*Sin*), and assignments (*Set*). All expressions are written within a function definition *name(args)*, which was defined by an initialization code declaration (*Init*) or by an evaluation code declaration (*Code*). The precise function body of each code term is determined automatically by a data flow analysis. This is controlled indirectly by the *level* numbers of the parameters a term depends upon and also by the *level* numbers of the assignments, which needs the term.

During reading of the term file, an internal term graph is created and constant expressions are evaluated immediately. After this, several modifications and simplifications are done as shown in Fig. 7. First, simplifications as $0 \cdot a = 0$, $1 \cdot a = a$ are performed. This is a very important step, because it can eliminate unneccessary expression trees, which occur by multiplication of matrices with zeros and other constant expressions. Due to this simplification, it is efficient to multiply a matrix by permutation matrices, selection matrices, and special rotation matrices. Other simplifications use commutative and associative properties of the underlying operators and collect sign changes by odd and even operators, leading to slightly simplified and unified expressions. It has to be noted that it is impossible to obtain the optimal simplified expression, but some heuristics like the ones above are quite useful in practice.

Then, identical constants, parameters, and operator terms are identified by fast hash tables and subexpressions are deleted, which are not used by later assignments.

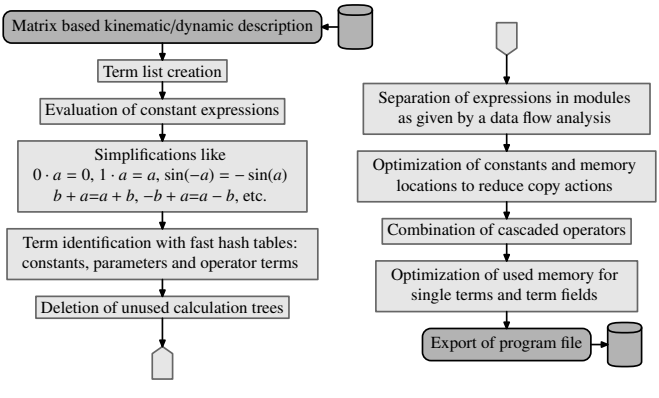

Matrix based kinematic/dynamic description
Term list creation
Evaluation of constant expressions
Simplifications like
$0 \cdot a = 0$, $1 \cdot a = a$, $\sin(-a) = -\sin(a)$
$b + a = a + b$, $-b + a = a - b$, etc.
Term identification with fast hash tables:
constants, parameters and operator terms
Deletion of unused calculation trees
Separation of expressions in modules
as given by a data flow analysis
Optimization of constants and memory
locations to reduce copy actions
Combination of cascaded operators
Optimization of used memory for
single terms and term fields
Export of program file

Fig. 7. Tasks of the symbolic processor *SymbolicCode*.

Table 3. Description of some operators in term files.

Term	Description
`Init` *level name(args)*	Function name and arguments for initialization code of priority *level*.
`Code` *level parent name(args)*	Function name and arguments for evaluation code number *level* with parent code number *parent*.
`Const` *value*	Constant floating point number *value*.
`Param` *name index level*	Parameter *name[index]* of initialization code or evaluation code *level*.
`Add` *arg1 arg2*	Summation operator $T_{arg1} + T_{arg2}$.
`Sub` *arg1 arg2*	Subtraction operator $T_{arg1} - T_{arg2}$.
`Mul` *arg1 arg2*	Multiplication operator $T_{arg1} \cdot T_{arg2}$.
`Div` *arg1 arg2*	Division operator T_{arg1}/T_{arg2}.
`Neg` *arg*	Sign change $-T_{arg}$.
`Cos` *arg*	Trigonometric operator $\cos(T_{arg})$.
`Sqrt` *arg*	Square root operator $\sqrt{T_{arg}}$.
`Set` *level param arg*	Assignment $T_{param} = T_{arg}$ for code *level*.

The data flow analysis now separates the terms into appropriate initialization/ evaluation code modules. Finally, memory locations for the constants are optimized to reduce copy actions as far as possible. Expressions like $sin(a + b * c)$ are identified, whose subterms are not used elsewhere. They are written in the target language format without the explicit use of memory locations for the temporary values $b * c$ and $a + b * c$, allowing the compiler to use processor registers more easily. Finally, a last optimization yields optimized memory locations for single terms and arrays of terms with respect to occupied memory. The program code is then written to a program target file in the specified language, which is C in most cases.

6.3 The Intermediate Tensor Scripting Language

In order to set up the relevant equations comfortably, a tensor language has been developed in the scripting language PERL. Each toolbox will have a specialized input syntax, but the tensor language is the underlying framework, in which the equations are formulated and constructed. This language uses lists of term numbers as the basic object representation. Every time a new constant (e.g., `$c = Constant 2.13`) or new parameters (e.g., `$p = NewParam "c_AS",3,3`, describing three center of gravity vectors) are defined, the corresponding low-level terms are written to the symbolic file, and their numbers are inserted in the corresponding sublists. Each basic operator has a corresponding tensor operator (e.g., `$res = Mult $a1,$a2` or `$res = Neg Plus $a1,$a2`), working element wise and expanding the symbolic file with corresponding single operator instructions, as shown in Table 3. In addition,

there are higher level operators like the tensor dot product, which is often used to multiply matrices (order two tensors) or to calculate the scalar product.

With this basic library, the application dependent parts can be expressed. For kinematic and dynamic equations, the concept of a moving coordinate frame as shown in Sect. 4 is used. They are implemented as objects, containing origin vectors $\boldsymbol{r}$ and orientation matrices $\boldsymbol{A}$, their derivatives $\dot{\boldsymbol{r}}$, $\dot{\boldsymbol{A}}$, and tensors to describe the accelerations $\ddot{\boldsymbol{r}}$, $\ddot{\boldsymbol{A}}$ and variations $\delta\boldsymbol{r}$ and $\delta\boldsymbol{A}$. For the initial world frame, the derivatives and variations are zero, and for the other constructed frames, they can be calculated by

$$\begin{aligned} \dot{\boldsymbol{r}} &= \boldsymbol{G}_{\boldsymbol{r}}(\dot{z}), & \dot{\boldsymbol{A}} &= \boldsymbol{G}_{\boldsymbol{A}}(\dot{z}), \\ \ddot{\boldsymbol{r}} &= \boldsymbol{G}_{\boldsymbol{r}}(\ddot{z}) + \hat{\boldsymbol{r}}, & \ddot{\boldsymbol{A}} &= \boldsymbol{G}_{\boldsymbol{A}}(\ddot{z}) + \hat{\boldsymbol{A}}, \quad \delta\boldsymbol{r} = \boldsymbol{G}_{\boldsymbol{r}}(\delta z), \quad \delta\boldsymbol{A} = \boldsymbol{G}_{\boldsymbol{A}}(\delta z). \end{aligned} \tag{9}$$

Here, z is the vector of generalized degrees of freedom of the structure, which has to be determined by the kinematic solver either in closed form by hand-written code or by numerical iterative solvers. One beautiful outcome of the hierarchical software architecture is the fact that more symbolic expressions can be generated in the frame library as needed by individual toolboxes. If some expressions are never used, they are silently erased by the low-level symbolic code processor. And by exporting expressions to different code functions, their subexpressions are automatically assigned to the correct calculation block. This permits the formulation of logically connected expressions at one place, even if they are evaluated at quite different places in the target code. Therefore, it is, for example, possible to describe revolute joints or translations once for all toolboxes by accumulating their effects on $\boldsymbol{r}$, $\boldsymbol{A}$, $\dot{\boldsymbol{r}}$, $\dot{\boldsymbol{A}}$, $\boldsymbol{G}_{\boldsymbol{r}}$, and $\boldsymbol{G}_{\boldsymbol{A}}$.

For each body in the traversal of chains, the inertial terms of eq. 2 are accumulated either in a global mass matrix $\boldsymbol{M}$ and a generalized force vector $\hat{\boldsymbol{f}}$ to solve for the accelerations (e.g., for time integration of the underlying differential algebraic equations) or more simple and efficient to sum up all generalized forces if the accelerations are explicitly given from the trajectory planner (e.g., for the inverse dynamic model). If only the exported equations of a general purpose multi-body system were used, such optimizations could not be done a posteriori. This can affect the processing time of the generated real-time code significantly.

7 Conclusion

General purpose multi-body systems are well suited to solve the kinematic and dynamic equations of parallel robot systems; but in practice it is often advantageous to have direct access to the underlying system equations as well as to the implemented data flow in the equations. Therefore, special tools, which can automatically generate these equations from a high level description of the robot structure are suggested. The tools have been testet by comparing the results of the created equations with results from the general purpose multi-body systems mentioned in the introduction.

The software hierarchy is established by three layers. The upper layer describes the problem in an application-oriented simple descriptive form. The intermediate layer is a matrix/tensor based scripting language currently implemented in PERL. The bottom layer consists of a very fast symbolic processor, which optimizes the equation tree of elementary terms with some sophisticated strategies producing C-code to be finally incorporated in the real-time environment or in simulation environments like MATLAB©. The overall procedure to generate kinematic and dynamic models was discussed in a general way. Special treatment was given to the creation of inverse dynamic models, which take a central role in the computed torque method to enhance the dynamics of a robot system by suitable control laws.

Acknowledgements. The author gratefully acknowledges the support of this work by the German Research Foundation (DFG) within the Collaborative Research Centre SFB 562 "Robots for Handling and Assembly — Highly Dynamic Parallel Structures with Adaptronic Components".

References

1. G. INTEC Gmbh (2010), `http://www.simpack.de` (accessed: January 10, 2010)
2. M. Software (2010), `http://www.adams.com` (accessed: January 10, 2010)
3. Bremer, H., Pfeiffer, F.: Elastische Mehrkörpersysteme. B.G. Teubner, Stuttgart, Germany (1992)
4. Schwertassek, R.: Dynamik flexibler Mehrkörpersysteme. Vieweg Verlag, Braunschweig (1999)
5. Eich-Soellner, E., Führer, C.: Numerical Methods in Multibody Dynamics, B.G. Teubner, Stuttgart, Germany (1998)
6. Rose, M., Sachau, D.: Multibody simulation of mechanism with distributed actuators on lightweight components. In: 8th Annual International Symposium on Smart Structures and Materials SPIE, Newport Beach, California (2001)
7. Rose, M., Sachau, D.: Multibody systems with distributed piezoelectric actors and sensors in flexible bodies. In: Design Engineering Technical Conferences ASME, Pittsburgh, USA, p. 11 (2001)
8. Rose, M., Keimer, R., Breitbach, E., Campanile, L.: Parallel robots with adaptronic components. Journal of Intelligent Material Systems and Structures 15, 763–769 (2004)
9. Budde, C., Last, P., Hesselbach, J.: Workspace enlargement of a triglide robot by changing working and assembly mode. In: Proc. of the IASTED International Conference on Robotics and Application, Cambridge, USA, pp. 244–248 (2005)
10. Rose, M., Keimer, R., Algermissen, S.: Vibration suppression on high speed parallel robots with adaptronic components. In: 10th International Congress on Sound and Vibration ICSV, Stockholm, Sweden, pp. 7–10 (2003)
11. Rose, M., Algermissen, S., Keimer, R., Budde, C.: Position dependent linearized elastic models of parallel robot systems with embedded smart devices for vibration suppression control algorithms. In: 14th International Congress on Sound and Vibration ICSV, Cairns, Australia (2007)
12. Algermissen, S., Rose, M., Keimer, R., Breitbach, E.: High-speed parallel robots with integrated vibration-suppression for handling and assembly. In: 11th Annual International Symposium on Smart Structures and Materials SPIE, San Diego, USA, p. 10 (2004)

13. Algermissen, S., Keimer, R., Rose, M., Breitbach, E.: Applied robust control for vibration suppression in parallel robots. In: 22nd International Symposium on Automation and Robotics in Construction ISARC, Ferrara, Italy (2005)
14. Rose, M., Algermissen, S., Keimer, R.: Automatically generated parallel robot system equations, based on a hierarchical software architecture. In: 9th International Conference on Computational Structures Technology CST, Athen, Greece (2008)
15. Rose, M.: Dynamische Modellierung der HEXA-Parallelstruktur. Technical Report IB 131-2004/13, German Aerospace Center (DLR) (2004)
16. Rose, M.: Dynamische Modellierung der TRIGLIDE-Parallelstruktur. Technical Report IB 131-2007/28, German Aerospace Center (DLR) (2007)
17. Hesselbach, J., Budde, C., Maaß, J., Breitbach, E., Rose, M.: Dynamic performance enhancement of a HEXA-parallel-robot using the computed torque approach. In: Proc. of Mechatronics & Robotics, vol. 3, pp. 1006–1011. Sascha Eysoldt Verlag, Aachen (2004)
18. Dietmaier, P.: The stewart-gough platform of general geometry can have 40 real postures. In: Advances in Robot Kinematics: Analysis and Control, pp. 1–10. Kluwer Academic Publishers, Dordrecht (1998)
19. Husty, M.: An algorithm for solving the direct kinematic of stewart-gough-type platforms. Mechanism and Machine Theory, 365–379 (1994)
20. Merlet, J.: Singular configurations of parallel manipulators and grassmann geometry. I.J. Robotic Res. 8(5), 45–56 (1989)
21. Chiaverini, S., Siciliano, B.: Closed-loop inverse kinematics algorith using quaternions. In: Proceddings of the 15th IMACS World Congress, vol. 1, pp. 741–746 (1997)
22. Thomas, U., Wahl, F.: A Unified Notation for Serial, Parallel and Hybrid Kinematic Structures. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 3–15. Springer, Heidelberg (2010)
23. Rauh, J., Schiehlen, W.: Various approaches for the modeling of flexible robot arms. In: Elishahoff, I., Irretier, H. (eds.) Refined Dynamical Theories of Beams, Plates and Shells and their Applications, pp. 420–429. Springer, Heidelberg (1987)

Knowledge-Based Design Principles and Tools for Parallel Robots

Carsten Stechert, Hans-Joachim Franke, and Thomas Vietor

Abstract. Complex products such as parallel robots need sophisticated management of knowledge and an integrated use of different tools during the whole product development process.

At the outset, this paper presents methods for knowledge structuring, such as object-oriented systems modeling and computer-aided design catalogs. It is shown how these approaches assist in early development phases (requirements management and goal conflict analysis), middle phases (concepts for modular products and reconfiguration) and later phases (definition of shape parameters and case based reasoning). In the end, the development environment demonstrates coherent data exchange and rule processing between the different design tools involved.

1 Introduction

The multidisciplinary development of complex products is often performed in collaborative networks. Hence, the system is broken down into smaller and manageable subsystems (or subtasks). Ideally, these subsystems can be handled independently. It is a major challenge to keep the subsystems consistent, because boundaries are diffuse and changes can have an effect on several subsystems at the same time. A sophisticated management of knowledge is needed and an integrated use of different methods and tools must be carried out through the whole product development process.

A goal-oriented approach should consider all important system views during the product lifecycle. Requirements are seen as the core of successful product development. Requirements sets must—as far as possible—be cleared of redundancies and inconsistencies, analyzed for goal conflicts, and distributed to involved

Carsten Stechert · Hans-Joachim Franke · Thomas Vietor
Technische Universität Braunschweig, Institute for Engineering Design,
Langer Kamp 8, 38106 Braunschweig, Germany
e-mail: {stechert,franke,vietor}@ikt.tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 59–75.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

stakeholders. The robot structure has to be developed following the strategies of modular systems to achieve economies of scale, to decrease time-to-market, and to provide reconfiguration possibilities. Finally, developers should be assisted by an integrated development environment during the whole robot development process.

On the following pages, this contribution will provide a brief overview of how the aforementioned aims can be attained in the field of parallel robots for handling and assembly.

2 Structuring Expert Knowledge

One considerable problem of methodical product development is the extreme growth of potential solutions with increasing concretization (combinatoric explosion [1]) and at the same time low decision certainty with regard to suitability and optimality in early design phases. This always leads to serious goal conflicts between a technologically optimum solution and acceptable time and cost schedules, especially in the field of complex mechatronic parallel robotic systems. It is therefore important to cut off branches of possible solutions in the solution tree as early as possible, so that development quickly progresses to phases of safe and certain (quantitative) design and optimization methods with a finite modeling effort.

It is possible to distinguish two basic principles:

1. Choosing the most suitable solution ideas from an appropriate selection (e.g., design catalogs of abstract solution principles, mechanisms or components) by applying task-specific selection criteria as early as possible.
2. Generating just those solutions (by combination or variation) that are generally applicable for a special task (method of conditional combination [2]).

The efficiency of the design process can be significantly increased by using knowledge-based design methods [3]. Most procedures for parallel robot development are affected by single technological problems (e.g., work space description, joint embodiment). Furthermore, development of multidisciplinary products needs methods for interdisciplinary information exchange and an abstract platform for collaboration [4]. Holistic approaches for an integrated development are absent.

In the following, it will be shown how the necessary knowledge for efficient conceptual and embodiment design, as well as for the use of parallel robotic systems with adaptronic components, can be gathered, systematically processed, and made available in computer-aided form.

2.1 Object-Oriented Systems Modeling

According to [5], a model should represent the subject to be modeled, ignore unimportant details (abstraction), and allow a pragmatic usage. Its purposes are to support and improve the understanding of the matter and to build a common basis for discussion and information exchange. Moreover, models should allow comparison of different solutions as well as analysis and prediction of the behavior and

characteristics of the system to be designed. The organization of a model should contain its structure and architecture. Furthermore, component internal dependencies, interactions between components and important external relations should be taken into account.

One important aspect of a model is its representation, especially the visualization of its contents for collaboration, as designers are seen as "visual thinkers" [6]. Salustri et al state two principles [7]:

- Simplicity is power.
- Diagrams augment cognition.

A model of the designed product is always a kind of documentation too. It documents the actual state of work, allows for discussions and provides a basis for presentations to stakeholders. The necessary degree of formalism depends on the project and its state. The more creativity is demanded, the more formalism would hinder. For instance, in the early phases a good designer would start with a free-hand sketch rather than using a CAD system for his very first ideas. On the other hand, in later phases a formalized workshop drawing or a detailed 3D-CAD model is necessary to exchange the generated information in a commonly understandable language and to transfer information to other models, e.g., FEM. Thus, many product development processes suggest a from-rough-to-detailed approach. The model should just be as detailed as necessary to provide a commonly understandable basis for all persons who might work with this model. Wherever more detailed aspects are needed, a submodel for a subset of project members should be generated.

The Systems Modeling Language (SysML) is an object-oriented approach to model a product on different levels of abstraction and with different viewpoints. It is a widely known notation within the fields of software development, electronic design, automation, and (in parts of) mechanical engineering. A variety of different (commercial) software tools is available. Here, SysML is used as an interdisciplinarily understandable notation for early modeling and knowledge structuring.

Within this project, SysML is extended to consider the following partial models of the early phases: goal system, product lifecycle, stakeholder network, product environment, project/product monitoring and system context. An extract will be shown in sections 3 and 4.

2.2 Computer-Aided Design Catalogs

Design catalogs in this context are able to transfer knowledge between development partners and to systematically inspire new and innovative solutions [8]. The basic tasks of design catalogs are to gather expert knowledge, to file it in a systematically structured way, and to make it available in a user-related form for a methodical development. A traditional static design catalog consists of four areas: classification part, main part, access part, and annex. Each part consists of a limited and

Fig. 1. Computer-aided design-catalog: screenshot of the structure tree and catalogs for joint concepts in an Internet browser.

pre-defined set of attributes. The attributes of the classification part span a consistent tree with every possible combination of attribute values. If a leaf of this tree cannot be matched with an existing solution, a *white field* is found that might be the starting point for a new class of solutions.

Design catalogs have two basic weak points, both of which are however solvable by using computers [8]. Firstly, the pre-definition of attributes in the classification part is just sufficient for the initial problem, because for clarity reasons it is useful to limit the classifying attributes to three or four. Hence, the spanned tree is not complete. For a different problem, different attributes should be used to structure the knowledge in a task-specific way and to eliminate unsuitable branches. Secondly, design information has to be up-to-date and accessible for every team member. Therefore, new information should be addable and the catalog should be accessible from each place independently of the computer platform used.

The actual design catalog software is based on .NET technology. It is accessible via standard Internet browsers from wherever Internet is available and allows intuitive use and flexible structuring with drag-and-drop and sorting rows. Apart from robot-specific catalogs, the system provides general information, such as physical effects or bionic principles. In the robotic directory, it provides catalogs for modules and interfaces of the modular system described in section 4.1. Moreover, systematic compositions of hybrid and fully parallel robotic systems, passive and active joints, characteristic kinematic chains, drives, active rods and sensors are provided. Fig. 1 shows a screenshot of the structure tree on the left side and catalogs for joint concepts in new windows of the Internet browser.

3 Managing Complex Requirements

Parallel robots can exist in a huge variety of different kinematic structures that provide widely differing characteristics. Hence, they are not equally suitable for different tasks. An effective choice of suitable structures is just possible with the help of systematic methods and tools. If a reasonable set of requirements was not considered during development, the parallel robot would not match the task and would provide too poor performance. However, it is difficult to handle requirements for the development of a complex mechatronic product. The huge number of requirements leads to problems in consistency, redundancy, transparency and actuality.

Fig. 2 gives a schematic overview of the different fields of input into the robot design process. Apart from customer wishes (characterized by specific use cases and goals during robot operation) and boundary conditions (such as available production processes), strategic aims of the different companies and departments involved have to be considered. The input data has to be systematically gathered, processed and made available as consistent sets of requirements for different abstraction levels and different components. The requirements sets can now be used for new development or to choose a suitable component from an existing database (cp. section 2.2 and section 4.3).

The processing of the requirements is a very important step not only to minimize the requirements to a consistent, non-redundant and structured set, but also to identify possible goal conflicts and change impact.

3.1 Systematically Structured Requirements

Achieving a structured database of requirements has two aspects. Firstly, basic requirements must be gathered by analyzing existing systems, e.g., with the help of

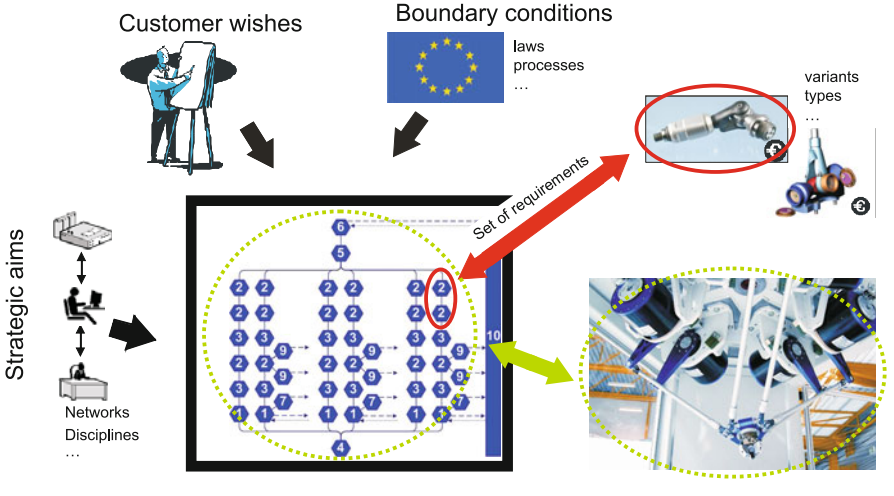


Fig. 2. Requirements and constraints for the HEXA-robot (schematic overview).

design catalogs of hybrid and fully parallel robots mentioned in section 2.2. Basic requirements result from typical structure and component attributes of parallel structures (e.g., lightweight construction) and the established and the conceivable use cases. Within the task field of handling and assembly, six task classes were identified with regard to the requirements degree of freedom (DOF) of the structure, payload, workspace shape and dimensions, accuracy and cycle times. The different requirements lists can thus be allocated to the hierarchical product structure considering product, assembly groups and components.

Secondly, a product specific terminology as commonly accepted vocabulary has to be developed. This allows a consistent formulation and a sound interpretation by all developers involved.

On an abstract level, requirements can be handled using a commercial SysML-tool [4]. It is possible to describe each requirement as an object, file it in an hierarchical structure (e.g., goal → target → requirement → subrequirement), and allocate it to one or—usually—more objects of the product structure (e.g., components). View-specific requirements lists, diagrams and matrices can be generated to clarify the observed problem. However, to model in SysML, a modeling software has to be installed on the client, and the client must be able to access the repository on the server.

As a more user-friendly solution, a requirements-tool has been developed [9]. It facilitates distributed collaboration as it uses the Internet with a platform-independent Java-Applet. The database consists of two coupled subdatabases: one for the terminology and the other one for the requirements lists. Users are able to regard existing lists, to build up new lists and to modify entries (if access permissions have been set) [10].

3.2 Relations of Requirements

As in every system model, objects (in this case requirements) are related to each other and to objects (abstract entities: e.g., parts, parasitic effects, use cases) of other partial models. In early phases, no concrete statements can normally be made, because the final relations depend on the final realization of the complete system.

Relations can be classified by what and how they relate to objects. These relation types are not mutually exclusive, e.g., a negative relation can be defined quantitatively. Table 1 shows seven identified types of relations in a condensed form. If the objects were not related following these relation types, they could be seen as independent, i.e., two independent objects having no impact on each other.

The first group of relations shown is allocated to the product development process. It starts with relations from goals to targets and ends with relations from requirements to test specifications. Granularity describes the hierarchy of related objects [11]. A system can be both an assembly of parts, and—more abstractly—a group of abstract objects like requirements, functions or use and test cases. Moreover, surrounding systems (e.g., component supply for automated assembly, weather conditions) have to be considered too. Relations can be considered to be directed,

Table 1. Types of relations concerning the requirements model (cp. [4]).

Development steps	**Granularity**	**Support**	**Linking**
goal-to-target	*system-to-system*	*positive*	*direct*
target-to-requirement	*part-to-part*	*negative*	*indirect*
requirement-to-requirement	*part-to-system*	*exclusionary*	
requirement-to-product-surrounding	**Direction**	**Quantifiability**	**Conditional**
requirement-to-component	*unidirectional*	*qualitative*	*if-then*
requirement-to-test-specification	*mutual*	*quantitative*	

i.e. the impact of the relation is just unidirectional (propagation blocker) or mutual. Support indicates whether there is a so-called goal conflict. A goal conflict occurs if requirements are related in such a way that their optimum realizations are contradictory (e.g., for a higher dynamic the robot accuracy drops). Quantifiability distinguishes two groups: those that can be expressed in a quantitative manner (i.e., a sophisticated equation can be stated to describe the relation) and those that can just be expressed qualitatively (e.g., the bigger the radius at the base-frame edges, the smoother it will appear in the eye of the beholder). The linking of two objects can be direct or indirect. In a chain of at least three elements, the first element directly affects its follower which in turn directly affects the third element. Although no direct linking was found, an indirect linking exists. The conditional relation describes that one object exists, because of the existence of another object, e.g., *if in operation, then provide low friction at sliding surfaces* (cp. [12] [13]).

In the requirements tool described earlier (cp. section 3.1), a functionality to handle relations was implemented [10] [14]. The considered requirements lists (i.e., considered components) span a triangular matrix. Now, an expert is able to mark direct relations as positive or negative. The tool is now able to find additional indirect relations. That way, goal conflicts so far unconsidered can be detected, and the impact a changed requirement can have on other requirements can be analyzed. The results are issued in table form or graphically in an HTML-file.

However, the tool is designed to aid developers from different domains and to exchange their knowledge. As a result, it is easy to use but does not provide all the functionalities a requirements analyst would need to process the requirements model. For a more elaborated modeling of requirements and relations, one possible approach is the extended SysML notation (cp. section 2.1). Fig. 3 shows a section of the model in order to illustrate the approach. On the right-hand side, use cases are shown as objects. One use case in the product lifecycle phase "Use" describes the handling of muffins. This use case leads amongst others to a refinement of the requirements *workspace* and *payload*. Besides fulfilling the specific use case *Handling of muffins*, one important goal is a *short cycle time*. There are a number of targets that support this goal. However, not every target is related to the development of the robot, i.e. is not within the project boundaries. The target *high dynamic* is directly related to the robot and supported by the requirements *high acceleration* and *high speed*. As a simplified example, the tripolar quantitative relationship between acceleration, speed, and workspace can be described by an equation considering constant

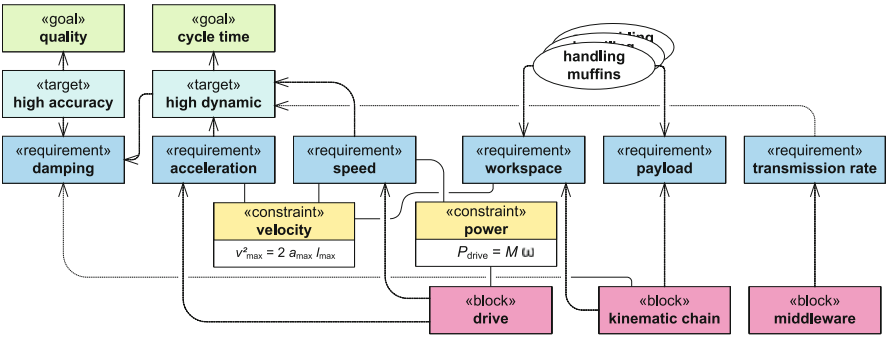


Fig. 3. Excerpt of the object-oriented requirements model (requirements diagram).

acceleration at the tool centre point (TCP). As long as the concretization level is low, this simplified equation can just give an idea of a reasonable area. However, it contains the danger of prejudgement. More information on this matter can be found in [15].

Where important relations were identified, more complex methods can be used. These are more linked up to physical reality than to expert experience. One approach is to find the relevant physical equations, to identify the input and output parameters as well as necessary empirical knowledge (e.g., material constants), and to relate the corresponding parameters [16]. Equations can be transformed into similarity ratios. The direction of optimization is integrated by assigning special parameters to nominator or denominator. Graph theory connects the ratios in such a way that the resulting graph describes the goal function of the subsystem and goal conflicts are identified on a physical basis [17].

4 Developing Modular Products

The development of modular systems is often considered where a number of different variants or a certain degree of flexibility is needed [18], i.e., modular systems are not only efficient for mass products but also for so-called mass customization products. The development of modular products is often an approach to save costs by economies of scale and to decrease time-to-market for a number of variants [19]. Hence, the number of internal variants has to be kept small, while the external variants should be high in order to fulfill additional customer wishes [19]. Trends in the field of robotics are systems that can be reconfigured while the robot is in operation [20]. The potential for static reconfiguration must be considered during modularization.

Drawbacks of modular systems are increasing development risks and development time. In addition, internal company processes and structures will become more complex. Thus, it is important to know the specific aims of the modular system development and to define modules in a proper way.

Around 40 more or less differing methods and methodologies to design modular systems are summarized in [19]. One common approach is based on three basic steps [21]:

1. Breaking down of the system into elements,
2. Documentation of the interactions between the elements, and
3. Clustering the elements into architectural and team chunks.

In the conceptual design phase, function structures are modeled that contain function elements and flows between them (e.g., [16]). In embodiment design, more detailed components are built up as function carriers and modules evolve.

In the field of parallel robots, some approaches for modular systems were taken for special structures or for just a part of the development process [22]. General approaches that consider requirements as a starting point and that accompany the development process for the definition of modules and interfaces cannot be found in literature. Such an approach has to be reproducible for the development of different kinematic structures for various use cases and surroundings. Apart from the kinematic structure, the base frame plays a fairly major role in the cost structure, but a modular system for it does not exist.

4.1 Kinematic Structures

The kinematic behavior of a parallel structure is mainly determined by the type and arrangement of joints in the kinematic chains. Hence, the structural synthesis [23] [24] gives a good starting point for the module definitions. The concept was developed as an open modular system, such that it can be used for a large part of known parallel kinematic structures, but also for as yet unknown structures. Then it was extended to additionally allow the configuration of hybrid structures. For a lifecycle-oriented modular system, models of different abstraction level and modeling profundity are needed [10] [14]. Starting with an abstract object-oriented model, a kinematic and dynamic model is evolved and, eventually a concrete CAD model is constructed.

The developed abstract structure scheme (cp. Fig. 4) contains ten modules shown as blocks in SysML notation (cp. section 2.1). Each block can be seen as a container that covers the characteristic parameters, e.g., length of a rod or DOF of a joint. A number of exactly defined interfaces exist between modules (shown as connecting lines). Thus, every structure can be made up of the structure scheme by arranging modules following the connecting lines. Modules can be used several times (i.e., loops can be repeatedly passed through) and connecting lines can be used in both directions. Not every module and interface has to be used for a concrete structure.

The blocks shown can be subdivided into submodules. Within the object-oriented model, relations and design rules can be represented. Following design rules, a concrete robot structure can be built up from rough to detailed. However, the more detailed the product model gets, the less useful an abstract model is, so that it is

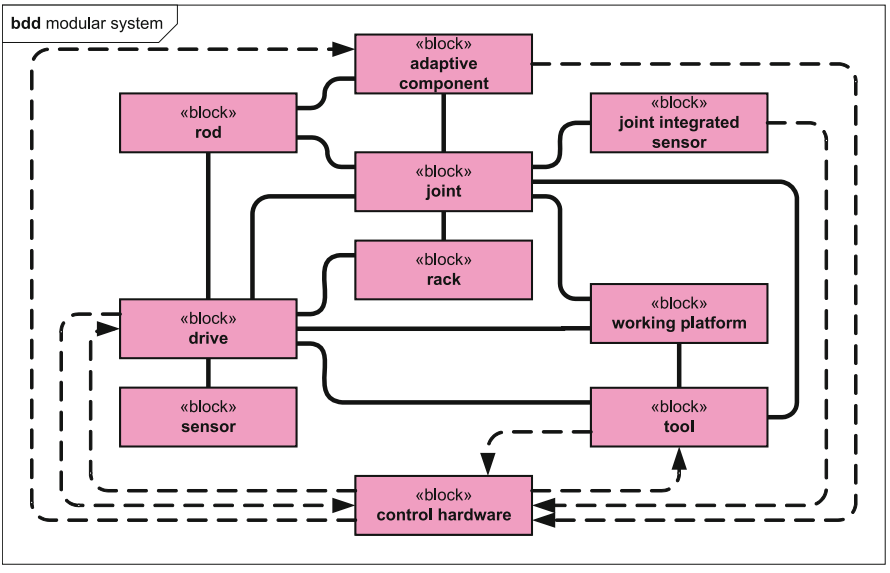


Fig. 4. Abstract structure plan of the modular system (— hardware interface, → data path).

necessary to switch to other models (e.g., CAD models). The abstract model stays present as the superordinate layer. In addition to being represented as objects in the model, the defined modules and interfaces are filed in the design catalog and made available to all team members (cp. section 2.2).

4.2 Base Frames and Racks

A robot base frame is often developed individually for one special robot and its specific surrounding situation, because the influence on stiffness, working and installation area and arrangement of drives is significant. The latter is also one important parameter for static reconfiguration. To date no standards and systematics for the development of base frames for parallel robots exist, although they are identified as one of the main cost-influencing components.

Within the aforementioned object-oriented model, the module "rack" is subdivided into submodules. The division was undertaken considering functional aspects and providing a large amount of flexibility and reusability of parts in case of reconfiguration. The modular system has to be an open system to allow the required adaptability to diverse surroundings. Easily available and already standardized extrusion profiles were identified as reasonable parts. It was found out that the purchase price of these semifinished parts is much higher than conventional parts. However, the resultant flexibility, a cheaper assembly and savings in transport due to the possibility of on-site assembly leads to a promising concept of modular base frames.

4.3 Requirements, Modules, and Case-Based Reasoning

As described earlier, requirements are collated in early phases of the development process and are allocated to system elements on different abstraction levels (cp. section 3). The requirements sets allow a step-by-step concretization of the product. On each level the requirements set is compared with parameters of the considered modules. If several concretizations are possible for one module, the designer can narrow the design space by choosing those solutions that best match the desired requirements. Eventually, concrete and already developed components can be chosen and assembled into a structure. If no matching component is found in the database (e.g., in the design catalogs of section 2.2), the requirements set is used to develop a new component or to modify one that matches quite well.

The described procedure shortens the development and manufacturing time and reduces the complexity and the number of intern variants. At the same time, quality is increased and risk is decreased, because most parts are already proven and tested.

5 Implementing the Development Environment

In the previous sections, it was shown how to structure knowledge and store it in design catalogs (section 2), generate good requirements sets for a given problem (section 3), and to use them together with a modular system to create new structures (section 4). However, qualitative rules and constraints alone are not able to create sound solutions for a specific robot task [25]. Simulations are necessary to verify solutions and to exclude possible—but unsatisfactory—variants. The available computer-aided tools (e.g., CAD, FEM, MBS) and specially adapted computer-aided methods for robot specific problems have to interact. This can be achieved by coupling of systems, integration or common system structures [3]. An integrated development environment needs to link comprehensive databases and to use standardized cross-system components [26]. At the same time, the system has to guide the user through the whole development process, assist the exchange of data, and help to avoid unnecessary iterations and useless regeneration of model data. Beyond that, different developers should be able to work simultaneously on the same project using the same actual dataset. A development environment must be able to ensure a consistent dataset for concurrent engineering.

In summary, the basic characteristics of an efficient development environment for parallel robots are the following:

- Flexibility thanks to a rule-based design,
- Integration of calculation, modeling, and simulation,
- Freedom in order of development steps, and
- Extendibility for new tools, elements, and rules.

5.1 Functionality

The actual development environment provides a variety of functionalities. Fig. 5 shows a use case diagram of the involved roles (stakeholders), tools and exchange

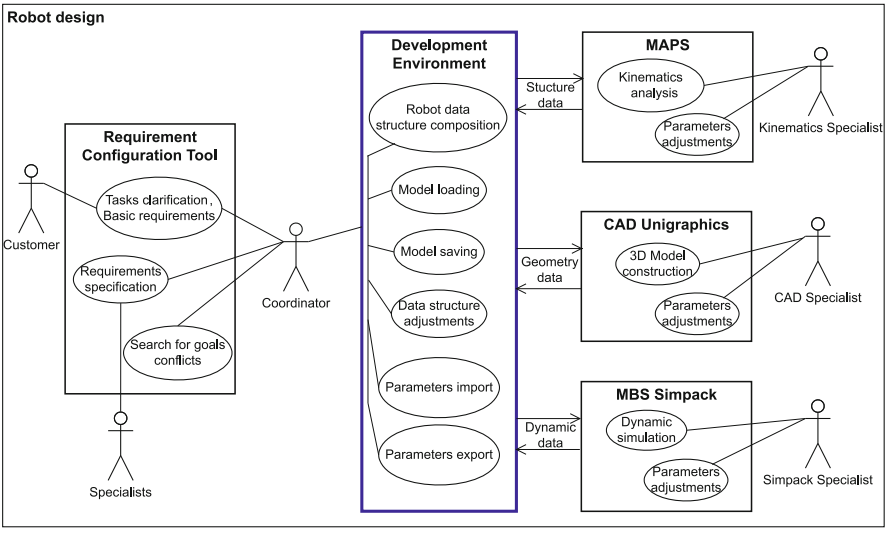


Fig. 5. Use case diagram of the functionalities of the actual development environment. Not shown is the interface to access the computer-aided design catalogs.

possibilities. The different roles are generally embodied by different persons, because the needed domain and tool-specific knowledge needed is often too complex, so that a single person cannot be an expert in all of these. The coordinator in the center is generally responsible for the project and coordinates the requirements configuration, the reasonable progress, and the work in the various domains.

The requirement configuration tool on the left side of Fig. 5 was already described in section 3.1. The coordinator is able to clarify the task and customer wishes to state basic requirements. Together with the specialists, the technical specifications are worked out. Afterwards, the coordinator searches (cross-system) goal conflicts and improves the requirements sets. Parallel to this, he uses the basic requirements for a reasonable structure synthesis. Following the selection matrix (after [23]) the desired DOF narrows down the possible solutions. Now the development environment assists the choice of the number of kinematic chains and a reasonable distribution of DOF for each chain and within a chain. As a first result, a possible kinematic structure is selected, e.g., a 6-RUS structure for six DOF. Then it is built up following the modular system's abstract structure (cp. section 4.1). Basic geometric data can now be added to provide the desired workspace, or default values can be used for a first loop.

At this point, model data can be saved or an earlier edited model can be loaded from the database. Furthermore, data can be exported to specialist tools. The first to be mentioned is MAPS—a Matlab based tool for kinematic analysis [23]. Here, the workspace characteristics (e.g., exact dimensions, position of singularities, possible orientations of the working platform) can be calculated based on the essential kinematic parameters (e.g., rod lengths, drive-fixture radius). If necessary, kinematic

parameters can be adjusted to fulfill the specific requirements. For instance, the drive-fixture radius can be decreased to provide a bigger workspace radius. The dataset thus changed can be imported back into the development environment. During the import process, the software checks the dataset. If it finds changes, it uses rules to recalculate those parameters not used in the kinematic analysis. The coordinator then has to approve all changes.

In a next step, data can be exchanged to the CAD system Unigraphics. Within Unigraphics, the model is built up out of prototype parts with simple geometries. It can be used as a virtual prototype, e.g., to visualize an early structure to the customer. This model has a relatively low complexity and can therefore be handled easily without exceeding the performance limits of conventional computer systems. Later in the design process, prototype parts can be replaced with finished CAD parts selected from the provided database (design catalog). If necessary, these parts can still be modified in the CAD system. In addition, with integrated CAD tools collision controls can be conducted for specific poses and physical characteristics can be determined. For instance, the CAD system can calculate mass and moments of inertia of joints and rods of a kinematic chain. These data can be used for the dynamic simulation.

The dynamic simulation is carried out in the Multi-Body-Simulation (MBS) Simpack. Here, it is possible to determine forces and moments that load the structure for special trajectories [27]. For instance, forces at the joints can be calculated. In a

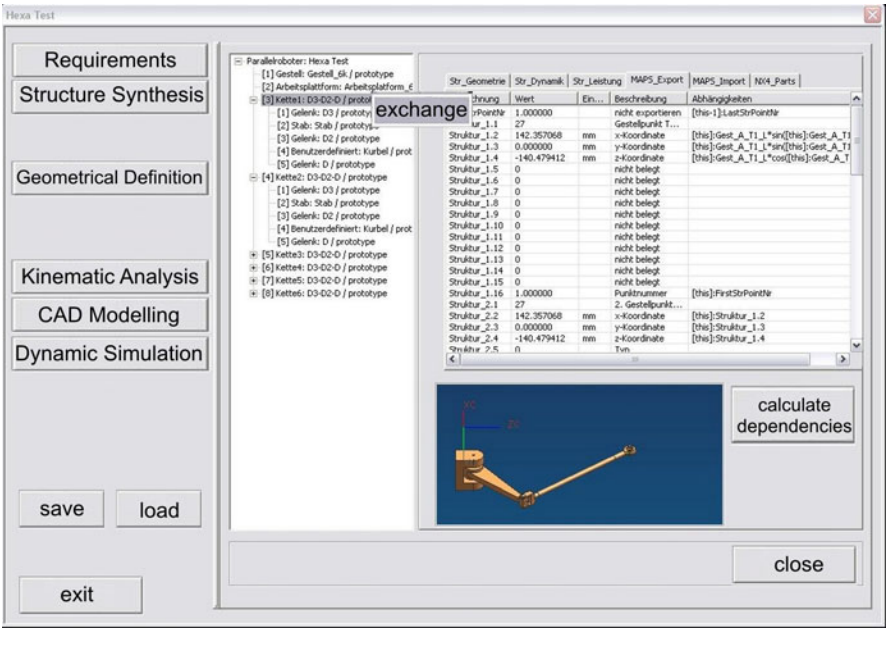


Fig. 6. Screenshot of the development environment with the MAPS-export tab open (data for kinematic analysis).

first iteration loop, they give a hint on choice of the right joints from the database. In a second loop, the simulation verifies whether the joints will withstand. The MBS interface is planned, but not yet implemented.

Fig. 6 shows a screenshot of the development environment. The buttons on the left-hand side start the different functionalities, e.g., structure synthesis. In the window center, the hierarchical model tree is located. With a right-click, prototype modules can be replaced with their finished counterparts. The right-hand side shows the model parameters. Different tabs provide special views for the different tools, e.g., MAPS. The table shown contains not only the parameters (name, value, and description), but also—if applicable—constraints to recalculate dependent parameters. Parameters as well as rules can be added and adjusted here.

5.2 Architecture

The development environment is realized in C++. Each component is described in its own xml-file. During structure synthesis, these files are read and merged, and dependencies are calculated. When saving the project, model data is stored in one structured xml-file. For every object, parameters are stored on the basis of categories (e.g., requirements, geometry). The parameter names are defined in such a way that they can be used unambiguously in every considered tool. As parts can be reused in the model (e.g., a HEXA-structure uses the same universal joint twelve times), data is stored in a hierarchical way, e.g., joints are subobjects of the kinematic chain. The superordinate object holds data that cannot be allocated to the subordinate part, e.g., position of a joint in the kinematic chain. Each time a new component is loaded, dependencies are refreshed.

One challenge was to provide movable robot structures in Unigraphics. The lower part of Fig. 7 shows the schematic chain model in the CAD system. The rods are connected to each other by constraints that depend on the degree of freedom and

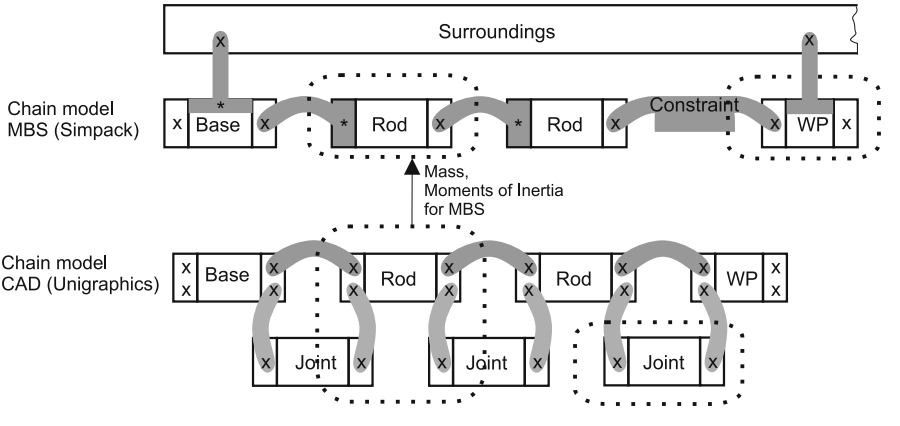


Fig. 7. Comparison of different chain models for CAD and MBS (WP - working platform).

possible orientation angles of the joint. Different joints and joints of different levels of detailing can be fixed to the rods. This is necessary to avoid structures containing subassemblies (e.g., joints), as due to level-spanning constraints the CAD system is unable to calculate possible movements.

In case of the dynamic model, the kinematic chain is made up as illustrated at the top of Fig. 7. Starting from the rack, the single kinematic chains are modeled as serial structures. Within the chains, rods are connected via constraints that simulate the joints. The upper ends of the chains are also connected via constraints to the working platform. As shown in Fig. 7, half of the mass of each real joint is allocated to the rod. Masses and moments of inertia are determined in the CAD-system and are used for MBS.

6 Conclusions

The last pages showed how to reach the aims that were set forth at the beginning of this contribution. A holistic process was described that starts at the early development phases with systematical gathering, processing and providing of requirements. For the middle phases, a modular system was introduced that allows a fast and sound generation of a wide variety of parallel structures. The process can be followed even to the late phases, where shape parameters are defined and the structure is verified by several simulations. The process is accompanied by a development environment and by knowledge-based databases. In addition, it is shown that an object-oriented modeling of the product helps to identify the most relevant requirements and relate them to other objects of the partial models concerned.

Acknowledgements. The authors gratefully thank the German research association (DFG) for supporting the Collaborative Research Centre SFB 562 “Robotic Systems for Handling and Assembly - High Dynamic Parallel Structures with Adaptronic Components”. This work is part of subproject A1 “Knowledge Based Development Methodology and Modular System”.

References

1. Franke, H.-J.: Design Methods before the Change of Paradigms. In: Grabowski, H. (ed.) Universal Design Theory, Aachen, Germany, pp. 249–269 (1998)
2. Kopp, F.: Ein Beitrag zur Struktursynthese von Mechanismen. Ph.D. thesis, Technische Universität Darmstadt (1973)
3. Adami, W.: Strukturen wissensbasierter Systeme für die rechnergestützte Konstruktion. Ph.D. thesis, Technische Universität Braunschweig (1992)
4. Stechert, C., Franke, H.-J.: Managing Requirements as the Core of Multi-Disciplinary Product Development. CIRP Journal of Manufacturing Science and Technology 1(3), 153–158 (2009)
5. Avgoustinov, N.: Modelling in Mechanical Engineering and Mechatronics: Towards Autonomous Intelligent Software Models. Springer, London (2007)

6. Lippardt, S.: Gezielte Förderung der Kreativität durch bildliche Produktmodelle. Ph.D. thesis, Technische Universität Braunschweig (2000)
7. Salustri, F., Eng, N., Weerasinghe, J.: Visualizing Information in the Early Stages of Engineering Design. Computer-Aided Design and Applications 5(5), 697–714 (2008)
8. Franke, H.-J., Löffler, S., Deimel, M.: Increasing the Efficiency of Design Catalogues by Using Modern Data Processing Technologies. In: Proc. of International Design Conference, Dubrovnik, Croatia, pp. 853–858 (2004)
9. Franke, H.-J., Otremba, R., Jänicke, T.: Methodical Development of Optimized Passive Joints. In: Proc. of the 1st International Colloquium of the Collaborative Research Centre 562: Robotic Systems for Handling and Assembly, Braunschweig, Germany, pp. 119–130 (2002)
10. Stechert, C., Wrege, C., Franke, H.-J.: Task-Based Modular Configurations for Hybrid and Redundant Parallel Robots. In: Proc. of 8th International IFAC Symposium on Robot Control, Bologna, Italy (2006)
11. Ariyo, O., Keller, R., Eckert, C., Clarkson, P.: Predicting Change Propagation on Different Levels of Granularity: An Algorithmic View. In: Proc. of 16th International Conference on Engineering Design, Paris, France (2007)
12. Stechert, C., Pavlovic, N., Franke, H.-J.: Parallel Robots with Adaptronic Components - Design Through Different Knowledge Domains. In: Proc. of 12th IFToMM World Congress, Besancon, France (2007)
13. Pavlović, N., Otremba, R., Inkermann, D., Franke, H.J., Vietor, T.: Passive and Adaptive Joints for Parallel Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 429–444. Springer, Heidelberg (2010)
14. Franke, H.-J., Wrege, C., Stechert, C., Pavlović, N.: Knowledge Based Development Environment. In: Proc. of 2nd International Colloquium of the Collaborative Research Center 562: Robotic Systems for Handling and Assembly, Braunschweig, Germany, pp. 221–236 (2005)
15. Stechert, C., Franke, H.-J.: Requirements Models for Collaborative Product Development. In: Proc. of 19th CIRP Design Conference, Cranfield, UK, pp. 24–31 (2009)
16. Franke, H.-J.: Untersuchungen zur Algorithmisierbarkeit des Konstruktionsprozesses. Ph.D. thesis, Technische Universität Braunschweig (1976)
17. Deimel, M.: Ähnlichkeitskennzahlen zur systematischen Synthese, Beurteilung und Optimierung von Konstruktionslösungen. Ph.D. thesis, Technische Universität Braunschweig (2007)
18. Liepert, B.: Industrierobotik, Quo vadis? In: Proc. of Robotik, München, Germany, pp. 1–2 (2004)
19. Firchau, N.: Variantenoptimierende Produktgestaltung. Ph.D. thesis, Technische Universität Braunschweig (2003)
20. Berhard, R.: Entwicklungstrends in der Robotik. ZWF - Zeitschrift für den wirtschaftlichen Fabrikbetrieb 95(3), 88–93 (2000)
21. Pimmler, T., Eppinger, S.: Integration analysis of product decompositions. In: Proc. of 6th International Conference on Design Theory and Methodology, Minneapolis, USA, pp. 343–351 (1994)
22. Pritschow, G., Wurst, K.-H.: LINAPOD - Ein Baukastensystem für Stabkinematiken. wt -Werkstatttechnik 87(9), 437–440 (1997)
23. Frindt, M.: Modulbasierte Synthese von Parallelstrukturen für Maschinen in der Produktionstechnik. Ph.D. thesis, Technische Universität Braunschweig (2001)
24. Frindt, M., Krefft, M., Hesselbach, J.: Structure and Type Synthesis of Parallel Manipulators. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 17–37. Springer, Heidelberg (2010)

25. Merlet, J.: The Necessity of Optimal Design for Parallel Machines and a Possible Certified Methodology. In: Proc. of 2nd International Colloquium of the Collaborative Research Center 562: Robotic Systems for Handling and Assembly, Braunschweig, Germany, pp. 7–20 (2005)
26. Gohritz, A.: Anforderungsbild an parallelstrukturgerechte Baugruppen. In: Proc. of Chemnitzer Parallelstruktur-Seminar, Zwickau, Germany, pp. 15–26 (1998)
27. Rose, M., Keimer, R., Breitbach, E., Campanile, L.: Parallel Robots with Adaptronic Components. Journal of Intelligent Material Systems and Structures 15(9-10), 763–769 (2004)

Detection and Avoidance of Singularities in Parallel Kinematic Machines

Franz Dietrich, Jochen Maaß, Carlos Bier, Ingo Pietsch,
Annika Raatz, and Jürgen Hesselbach

Abstract. In this work a geometrical and a physically based method for the detection of singularities are presented, which provide information about the distance of a given position to singularities. The integration of this singularity detection into a robot controller is presented and validated by experimental results. Based on these results a path planning algorithm that avoids singularities is developed. It uses a particle based randomizing scheme for finding a path within the workspace to which a virtual potential field is applied. The HEXAII demonstrator of the Collaborative Research Center 562 serves as a validation platform for this algorithm. The singularity avoidance path planner is integrated into the real-time context of the control application *RCA562*, from which experimental results are presented.

1 Introduction

Generic task-oriented descriptions of robot tasks, e.g. skill primitives, proved to be a powerful tool for efficient programming of robot movements. This concept

Franz Dietrich · Annika Raatz · Jürgen Hesselbach
Technische Universität Braunschweig, Institute of Production Automation and Machine Tools, Langer Kamp 19b, 38106 Braunschweig, Germany
e-mail: {f.dietrich,a.raatz,j.hesselbach}@tu-bs.de

Jochen Maaß
SkySails GmbH & Co. KG, Veritaskai 3, 21079 Hamburg, Germany
e-mail: j.maass@tu-bs.de

Carlos Bier
IAV GmbH, Rockwellstraße 16, 38518 Gifhorn, Germany
e-mail: carlos.cezar.bier@iav.de

Ingo Pietsch
BSH GmbH, Carl-Wery-Straße 34, 81739 München, Germany
e-mail: ingo.pietsch@bshg.com

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 77–92.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

allows tasks to be programmed without profound knowledge of a particular robot kinematic structure. The implementation on a parallel kinematic structure without the consideration of the structural properties might cause unintended or dangerous situations. For example, the robot may get trapped in a kinematic singularity of type 2. In such a pose forces within the structure tend to infinity. This results in unstable control loops or even structural damage. However, the paradigm of the generic description of robot tasks would be violated if structural properties of a particular robot would be considered in the robot programming interface.

This dilemma becomes even more important when looking at parallel robots that are especially designed for assembly and handling tasks. Their kinematics are usually optimized with respect to maximum spatial orientability. Unfortunately this implies that kinematic singularities are present inside the workspace in many cases [1, 2]. In consequence it is quite common to design the robot in such a way that wide areas of the workspace are free of singularities [3]. Alternatively, the kinematic blocks certain degrees of freedom [4], for example the spatial orientability. The drawback of this method is the loss of universality of the robot. This illustrates the conflict of objectives that structural and geometrical synthesis faces.

Thus, when working with task-oriented movement descriptions a method is required which allows robots with high spatial orientability to be moved safely and efficiently. Prior to the development of such planning methods an algorithm is required that detects the singularities to be avoided. This method should not only detect situations in which the mechanism has already reached the singularity, which would be too late anyway. It should also give a measure for the distance of a pose to singular poses.

The following section introduces singularity analysis based on Grassmann geometry. Here all the direct kinematic singularities of parallel robots are associated with one or more Grassman varieties. Based on this theory it is possible to obtain a closeness index as a distance measure between geometric elements (planes and lines). This distance measure is then mapped to represent the proximity of the parallel robot to a singularity. In the subsequent section a singularity analysis that relies on physical motivation is presented. This is considered as a powerful alternative to the Grassmann based singularity analysis. Here a proximity measure is obtained, which offers the great advantage of physically consistent measures of singularity proximity.

The physically motivated index type provides the fundament for path planning. For this problem an algorithm is proposed that relies on this index and its gradient. After introducing the base algorithm several ehancements are presented which shorten the time of convergence and feature recovery from dead-end situations.

2 Detection of Singularities

The methods for the analysis of singularities presented in this chapter involve the screw theory. Therefore the most important relationships, which are necessary for the understanding of the following sections, are given at first. The screw theory

and Grassmann's theory are used to elaborate geometrical analysis of singularities. Afterwards a physically inspired method for the analysis of singularities is presented, from which a closeness index is obtained. This method allows the physical meaning of the index to be chosen. It may represent the transmittable power, stiffness, kinetic energy or eigenfrequency.

2.1 Screw Theory

A screw $\$$ is a geometric element consisting of a directed line and a scalar length parameter called pitch [5]. If the directed line is represented by a normalized vector, the screw is called a normalized screw $\hat{\$}$. Any screw may be decomposed into a normalized screw and its magnitude Ψ [6], i.e. $\$ = \hat{\$}\Psi$. More literature on screw theory and its application may be found in [5, 7, 6, 8]. We also refer to [9], where a practical overview of screw theory is given under consideration of its applications to parallel mechanisms.

2.1.1 Transmitted Power Expressed by Screws

Consider a rigid body moving along an instantaneous twist $\$_t$, to which a wrench $\$'$ is applied ($\$'$ in axis order, $\$_t$ in ray order as denoted in [6]):

$$\$_t = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{V_p} \end{bmatrix}, \quad \$' = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{C}_0 \end{bmatrix} \tag{1}$$

The power transmitted by this movement is [5, 6]

$$\delta W = \boldsymbol{C}_O \cdot \boldsymbol{\omega} + \boldsymbol{f} = \$'^T \$_t. \tag{2}$$

2.2 Differential Kinematics of Parallel Manipulators

This section shortly describes differential kinematics of parallel manipulators for a better understanding of the following classification of singularities.

2.2.1 Differential Kinematics Based on Closed Vector Chains

For the description of the kinematics of serial manipulators recursive Denavit-Hartenberg notations are widely in use. In contrast, the kinematics of parallel manipulators are often described using geometrical methods with a closed chain of vectors. This method obtains a set of equations that is non-redundant, nonlinear and implicit [10, 11]:

$$\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{q}) = \boldsymbol{0}. \tag{3}$$

$\boldsymbol{q} = [q_1 \dots q_n]^T$ Drive coordinates
$\boldsymbol{x} = [X, Y, Z, \psi, \theta, \phi]^T$ End effector position and orientation
Differentiation of (3) with respect to time yields

$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} - \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} = \boldsymbol{J}_{xA}\dot{\boldsymbol{x}} - \boldsymbol{J}_q\dot{\boldsymbol{q}} = \boldsymbol{0} \tag{4}$$

$\dot{\boldsymbol{q}} = [\dot{q}_1 \dots \dot{q}_n]^T$ Drive velocities
$\dot{\boldsymbol{x}} = [\boldsymbol{v}^T, \dot{\boldsymbol{\Phi}}^T]$ Translational and angular velocities of end effector

It must be considered that the analytical forward Jacobian $\boldsymbol{J}_{xA}$ must be replaced by the geometrical forward Jacobian $\boldsymbol{J}_x$. If $\boldsymbol{J}_q$ is invertible, the combined Jacobian $\boldsymbol{J}$ may be computed from (4) [12, 13]. It is valid as long as the mechanism is in non-singular configurations [14]. With the notation $\$_t = [\boldsymbol{v}, \boldsymbol{\omega}]^T$ Equation (5) is obtained:

$$\dot{\boldsymbol{q}} = \left(\boldsymbol{J}_q^{-1}\boldsymbol{J}_x\right)\$_t = \boldsymbol{J}\$_t. \tag{5}$$

2.3 Singularities of Parallel Mechanisms

2.3.1 Classification of Singularities with Parallel Mechanisms

A general classification of singularities of parallel mechanisms was first introduced by Gosselin and Angeles in the late 1980ies [12]. This classification is based on the kinematic properties of the Jacobians shown in (5). A Jacobian is singular by definition if not all its partial differentials are linearly independent. Then the rank of the respective matrix decreases. With parallel kinematics three types of singularities may occur: Either $\boldsymbol{J}_q$ looses full rank (Type 1), $\boldsymbol{J}_x$ looses full rank (Type 2), or the both (Type 3)

Type 1: Type 1 singularities occur at points where the forward Jacobian becomes singular ($det(\boldsymbol{J}_q) = 0$). For a parallel manipulator this means that at least one drive may travel an infinitesimal distance while the position and orientation of the end effector will not change. In this configuration the end effector looses at least one degree of freedom. Singularities of type 1 are placed at the outer surface of the workspace, at positions where at least one limb is completeley straightened.

From the kinematic point of view these configurations are uncomplicated. The manipulator can easily reach these places and leave them again with its built-in drives. But to successfully control the manipulator in proximity of such situations the trajectory planning must be carried out in drive joint coordinates. Planning in cartesian space may generate velocities which unintentionally tend to infinity because of the lost degree of freedom of the transformation from end-effector coordinates to joint coordinates.

Type 2: At points where the determinant of the inverse Jacobian becomes zero a singularity of type 2 occurs ($det(\boldsymbol{J}_x) = 0$). Theoretically this means that the end effector may move an infinitesimal distance while the drives remain blocked at their positions. There the structure gains one or more unintended degrees of freedom. Beside the fact that multiple solutions of the forward kinematic problem may be found at this point the stiffness of the end effector decays towards zero. This means that the mechanism will not be able to move out of these positions

by itself. Consequently singularities of type 2 must be avoided. Singularities of type 2 may be located inside the workspace, but still they represent an inner border where two configuration spaces touch each other. This increases the importance of special control functionalities that monitor or even actively avoid type 2 singularities in parallel mechanisms.

Type 3: A singularity of type 3 occurs where both, the forward Jacobian and the inverse Jacobian become singular ($det(\boldsymbol{J}_q) = 0$ and $det(\boldsymbol{J}_x) = 0$). Therefore the properties of these configurations are already described by the combination of type 1 and type 2 singularities. Hence, singularities of type 3 will not be treated explicitly in the following.

2.3.2 Problems Caused by Singularities

When a parallel kinematic approaches a singular configuration, three problems occur. First, the accuracy decays due to the disadvantageous transmition ratio from joint coordinates to end effector coordinates. Second, inner forces acting in the joints may tend towards infinity. Third, the computation of forward kinematics may find and choose misleading solutions of the real configuration of the mechanism.

Accuracy: The combined Jacobian developed in (5) not only shows the transmission ratio of the velocities of the actuated joints and the end effector. It can also be transformed to reveal the transmission of inaccuracy:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}\$_t \quad \Rightarrow \quad \frac{\Delta \boldsymbol{q}}{\Delta t} \approx \boldsymbol{J}\frac{\Delta \boldsymbol{x}}{\Delta t} \quad \Rightarrow \quad \Delta \boldsymbol{x} \approx \boldsymbol{J}^{-1}\Delta \boldsymbol{q}. \tag{6}$$

This formulation shows how inaccuracy at the actuated joints $\Delta \boldsymbol{q}$ is transmitted to the end effector. Near singularities the transmition ratios of the inverse of the combined Jacobian $\boldsymbol{J}^{-1}$ increase drastically. This means that the repeat accuracy degrades considerably near singularities.

Inner forces: Similarly to the transmission of inaccuracies the transmission of forces increases when the mechanism approaches a singularity. The closer the mechanisms gets to a singularity the higher the forces the joints must transmit increase. Theoretically the forces tend to infinity when the mechanism approaches a singularity. This means that joints and bars must be designed to withstand this load.

Ambiguity of forward kinematics: In spatial parallel mechanisms the end effector position is usually computed by iteratively searching a valid solution of the forward kinematics. Unfortunately one set of joint coordinates may have multiple solutions of the end effector. These solutions may be located in distinct configuration spaces. If the initial guess is far away from the solution of the desired configuration space the search algorithm may slip through the singularity and tend towards another solution.

The selection of the wrong solution of the forward kinematics may also be caused by dynamic effects: When the end effector is located near a singularity it is possible that the mechanism crosses the singularity unintentionally due to

dynamic effects. In this way the mechanism switches to another configuration space and moves apart from the computed solution of the end effector position. This situation may cause loss of control or even structural damage.

2.3.3 Linear Algebra Singularity Analysis

In order to detect singularities the determinant of the Jacobian is often used [15]:

$$\kappa_M = ||\det \boldsymbol{J}|| \tag{7}$$

If the index becomes zero, a singularity is reached. The definition of a "Quality Index" κ_Q [16] is an extension to this strategy, which introduces normalization

$$\kappa_Q = \frac{||\det \boldsymbol{J}||}{||\det \boldsymbol{J}||_{max}}. \tag{8}$$

These methods provide power based indexes which are related to the smallest eigenvalue of the Jacobian. A small eigenvalue means that high forces must be transmitted by the passive joints and links. This is interpreted in such a way that the end effector is near a singularity. Unfortunately the indexes (7) and (8) do not have a geometrical representation. Additionally the definition of a safety distance fails because there is no lower limit which is valid across the whole workspace [17]. This is a consequence of the fact that the determinant consists of the product of the corresponding eigenvalues: When one eigenvalue tends towards zero it is likely that another one increases even faster and compensates the tendency towards zero. In this case the determinant will not reveal the closeness at all until it suddenly drops to zero.

Another proximity measure involving the Jacobian is based on the ratio of the smallest and the biggest eigenvalue of $\boldsymbol{J}^T\boldsymbol{J}$ [18]:

$$\kappa_K = \sqrt{\frac{\lambda_{\max}(\boldsymbol{J}^T\boldsymbol{J})}{\lambda_{\min}(\boldsymbol{J}^T\boldsymbol{J})}} = \frac{\sigma_{\max}(\boldsymbol{J})}{\sigma_{\min}(\boldsymbol{J})} \tag{9}$$

This method does not show the disadvantages of (7) and (8), because κ_K is independent of $\det \boldsymbol{J}$ [17]. Unfortunately this method is neither unit consistent nor independent of a specific coordinate system because the entries in the Jacobian are not consistent [19, 20, 21].

There exist some more ideas to treat Jacobians in such a way that a consistent measure can be found [22]. Unfortunately these methods also suffer from not being physically meaningful. This situation motivates our proposition of singularity indexes, which are elaborated in the following sections.

2.3.4 Geometrical Singularity Analysis (Grassmann Theory)

In most parallel manipulators the wrenches $\$_r$ only apply forces to the end effector platform (represented by null pitch screws) and their screw components correspond to Plücker components of a line, i.e. a Plücker line. Then the singular configurations

represent configurations where the wrenches contain a set of linearly dependent lines. They are called line-based singularities, where the end effector is able to move along an uncontrollable twist $\$_{tEE}$. Grassmann studied the varieties of lines, i.e. the sets of linear dependent lines to n given independent lines, and characterized them geometrically. The linear varieties are classified according to their rank.

Rank 1: The variety of rank one contains a line (wrench) in $\Re^3$. If any other line is linearly dependent on this line, the system is in a singularity.

Rank 2: The variety of rank two is composed of either a pair of skew lines (case 2a), or a flat pencil of lines (case 2b). Two skew lines cannot generate a third line which is linearly dependent on both of them, so it is not necessary to consider this case. Case 2b contains case 2b1, where the two lines intersect at a finite point, and case 2b2, where two lines are parallel (intersection at infinity).

Rank 3: The variety of rank three has four types. In case 3a the lines involved form a regulus. In case 3b three lines intersect in the same point and generate a bundle of lines. This case becomes the special case 3b1 if all lines intersect in a finite point. Special case 3b2 is given if the lines intersect at infinity. Case 3c consists of two flat pencils of lines having different centers located in different planes. There three variations can be distinguished. Case 3d is generated if three lines are located in the same plane.

Rank 4: This variety, called linear congruence, occurs when one or more lines are linearly dependent to four linear independent lines. There are four cases: Case 4a is an elliptic linear congruence where all lines of the variety form concentric hyperboloids related to a common axis. Case 4b, called hyperbolic congruence, occurs if all lines intersect simultaneously two imaginary skew lines. Case 4c (parabolic linear congruence) consists of two families of flat pencils sharing one line. This case is already covered by case 2b and case 3a and does not have to be considered.

Rank 5: This variety is called complexes and has two types: “general complex” and “degenerate complex”. A general complex (case 5a) contains all flat pencils whose center is in the helix and whose plane is normal to the helix tangent line in this point. This helix is generated by five skew lines. A degenerate complex (case 5b) is composed by all the lines meeting one given line. This case is subdivided depending on whether this line is at infinity (case 5b2) or not (case 5b1).

Figure 1 shows various parallel robots in singular configurations. The wrenches are also drawn to illustrate how Grassmann’s theory can be applied to parallel robots. Using the Grassmann geometry the cases of the singular configurations of a parallel manipulator may be identified. Each case may be characterized by geometrical constraints. Hence the problem of calculating the closeness of a position to a singularity can be reduced to the calculation of distances between straights and/or planes. The result is physically meaningful. However, if multiple singularities are located within the workspace multiple indexes must be joined to obtain a scalar value that represents the distance to the nearest singularity.

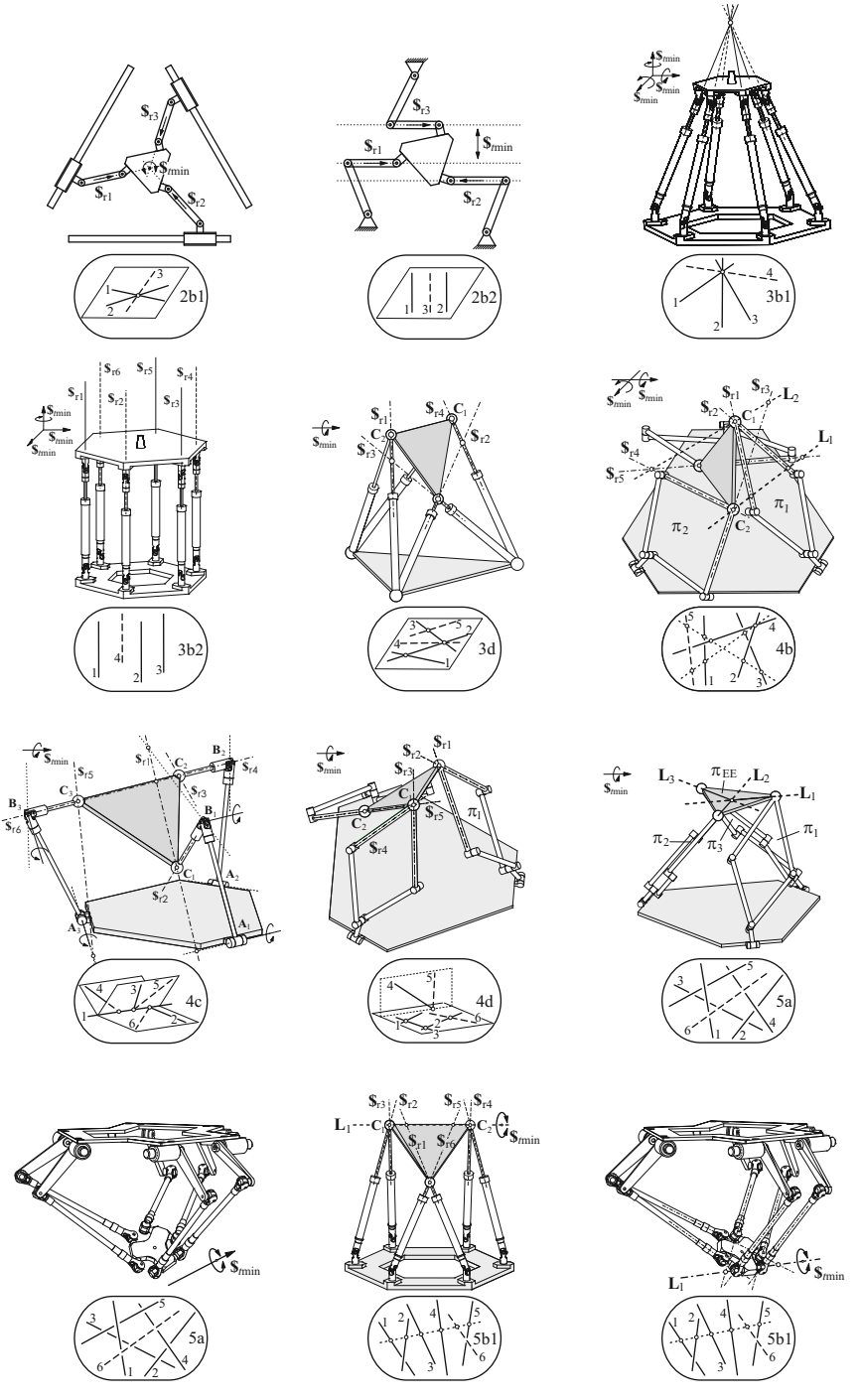


Fig. 1. Examples for Grassmann singularities on parallel robots.

2.3.5 Physically Based Singularity Analysis

In singularities but also in the proximity of these configurations physical properties change in an unintended way. For example, when the mechanism approaches a singularity the power transmitted from the drives to the end effector decreases down to zero. In the same way, when stiffness decreases the eigenfrequency decreases. Based on this idea a target function is sought that delivers one of these measures (transmittable power, direction of lowest end effector stiffness, lowest eigenfrequency) dependent on the pose of the end effector.

A general class of functions which deliver such indexes is described by (10). There the optimum of a scalar target function Zf represents the index value sought. The optimiziation is constraint by $\boldsymbol{Zb}_1$ and $\boldsymbol{Zb}_2$. $\boldsymbol{Zf}$ may be interpreted as an input to the system, $\boldsymbol{Zb}_2$ then is an output and $\boldsymbol{Zb}_1$ ties input and output together, i.e. via the kinematic or static transfer function.

$$M(\boldsymbol{x}) = \begin{cases} \text{min / max} & Zf(\dot{\boldsymbol{q}}) = \dot{\boldsymbol{q}}^T \boldsymbol{S} \dot{\boldsymbol{q}} & \text{(Input function)} \\ \text{constraints:} & \boldsymbol{Zb}_1 = \dot{\boldsymbol{q}} - \boldsymbol{J}\$_t = \boldsymbol{0} & \text{(Transfer function)} \\ & Zb_2 = \$_t^T \boldsymbol{T} \$_t - c = 0 & \text{(Output function)} \end{cases} \tag{10}$$

Using (5), $c = 1$ due to normalization, and combining $\boldsymbol{Zb}_1$ with Zf 10 may be rewritten:

$$M(\boldsymbol{x}) = \begin{cases} \text{min / max} & Zf(\$_t) = \$_t^T \boldsymbol{J}^T \boldsymbol{S} \boldsymbol{J} \$_t \\ \text{constraint:} & Zb = \$_t^T \boldsymbol{T} \$_t - 1 = 0 \end{cases}. \tag{11}$$

The optimum of $M(\boldsymbol{x})$ satisfies the relation

$$\det\left(\boldsymbol{J}^T \boldsymbol{S} \boldsymbol{T} - \lambda \boldsymbol{T}\right) = 0, \tag{12}$$

which represents a generalized eigenvalue problem with eigenvalues λ_i and their corresponding eigenvectors $\$_{ti}$. $\lambda_{\text{min/max}}$ represents the minimum / maximum of Zf while the twist $\$_{t,\text{min/max}}$ describes the corresponding movement of the end effector.

Particular physical meanings can be assigned to M by suitable choices of the weighing matrices $\boldsymbol{T}$ and $\boldsymbol{S}$ [9]:

Power / Stiffness: Considering the HEXAII the choice of $\boldsymbol{J} = \boldsymbol{J}_x$, $\boldsymbol{S} = \boldsymbol{I}_{6\times6}$ and $\boldsymbol{T} = \text{diag}[\boldsymbol{0}_{3\times1}, \boldsymbol{1}_{3\times1}]$ is reasonable in order to receive a proportional relationship between the eigenvalues and the transmittable power of each limb connected to the end effector ($\sqrt{\lambda_i} \sim W$). Alternatively, choosing $\boldsymbol{S} = \boldsymbol{K}_{\text{drive}}$ (where $\boldsymbol{K}_{\text{drive}}$ contains the drive stiffnesses) relates the λ_i to the end effector stiffnesses. If drive stiffnesses are equal (which is the case for the HEXAII and many other manipulators), the stiffness based measure is proportional to the power based measure. Therefore both approaches are considered to be interchangeable in this example. λ_{min} and its corresponding eigenvector contain the direction of minimal stiffness. When the transmittable power / end-effector stiffness reaches 0 the mechanism is in a singular configuration. The eigenvector shows the degree of freedom

which the mechanism gains. Outside of the singularities the twist eigenvector corresponding to the lowest transmittable power or lowest stiffness $\lambda_{\min}$ represents the direction of the lowest stiffness.

Eigenfrequency: When selecting $\boldsymbol{S} = \boldsymbol{K}_{\text{drive}}$ and $\boldsymbol{T} = \boldsymbol{M}_{\text{EE}}$ (where $\boldsymbol{M}_{\text{EE}}$ is the mass matrix w.r.t the end effector coordinates) the solutions are related to the squared eigenfrequencies ($\lambda_i \sim \omega_i^2$). The smallest eigenvalue $\lambda_{\min}$ and its corresponding eigenvector represent the smallest eigenfrequency in a given pose. The smallest eigenfrequency decreases steadily when the mechanism approaches a singularity. It becomes zero at a singular configuration. Again the corresponding eigenvector represents the additional degree of freedom the mechanism gains in direct proximity of the singularity.

3 Example: Singularity Supervision and Automated Recovery Strategy of the HEXAII Parallel Robot

In order to realize a workspace supervision of a parallel robot it must be checked for inner collision and singularities (type 1 and type 2). Here, especially the supervision of singularities of type 2 is of interest. The prerequisite for the realization of the supervision is that one single variable is able to monitor the distance to all singular configurations. This is met by the power based and the eigenfrequency based method presented in the above section. It provides a measure which consistently falls under a global constant limit when the mechanism is located near a singularity.

This monitoring algorithm was implemented in the HEXAII control application. Figure 2 shows experimental results based on the power inspired method. The limit below which the HEXAII enters the singular region and is not controllable anymore was determined experimentally. It can be seen that the measure index is approximately proportional to the distance of the end effector to the singularity, as desired. More experimental results recorded during moving sequences when singularities are passed can be found in [9, 23].

The workspace supervision stops ongoing movements instantly when the singularity proxmity measure falls below a predefined limit. Based hereon an automatic strategy to recover from areas which are close to singular configurations was developed. The robot moves along the gradient away from the singularity until the proximity measure re-enters the valid interval above the limit mentioned above again [24]. The robot is then safe to be moved in any direction again.

4 Path Planning Methods for Singularity Avoidance Using Particles

4.1 Motivation and Requirements

We already pointed out the demand for a planning algorithm that can be operated under realtime constraints and that is capable of finding a path free of singularities.

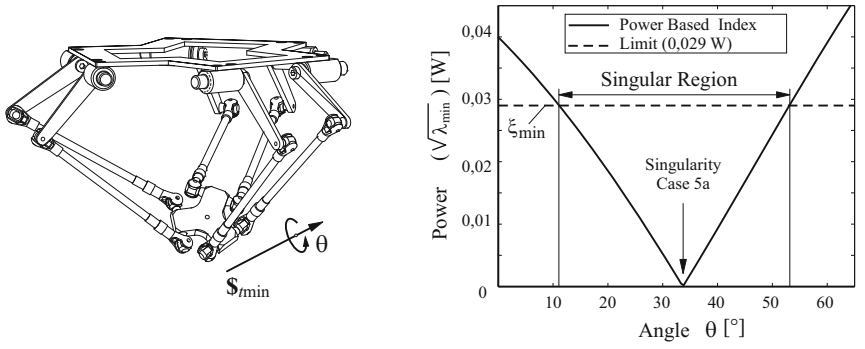


Fig. 2. Singularity distance measure close to a singularity of type 2: experimental results from power inspired method.

The application of an algorithm based on particles is appealing because there the singularity index has to be evaluated for discrete poses only. Here we understand a particle as a point in the work space and the corresponding singularity index. If required, the gradient with respect to the dimensions of the workspace may be attached optionally. It is desirable that the path is planned with minimum deviation from the shortest connection along predefined poses.

Since the singularity index can only be obtained numerically the computation of the gradient has to be numerical as well. Therefore, the gradient function has to be smooth. For the HEXAII robot, which serves as a validation example here, this prerequisite is sufficiently met [25, 26]. The algorithm presented in the subequent section was integrated into *RCA562*[1] [27]. A comparison of this algorithm versus other methods is contained in [25].

4.1.1 Preparation

The translational and rotational elements of the pose vector $\boldsymbol{x}$ have to be normalized first because the workspace of the robot is not a vector space. For this reason the weights ν are introduced:

$$\boldsymbol{N} = \mathrm{diag}[\nu_x \nu_y \nu_z \nu_{\phi_x} \nu_{\phi_y} \nu_{\phi_z}] \tag{13}$$

$$\check{\boldsymbol{x}} = \boldsymbol{N}\boldsymbol{x} \tag{14}$$

Subsequently a norm is define wich yields a vector space:

$$\|\boldsymbol{x}\|_{\mathrm{N}} = \sqrt{\check{\boldsymbol{x}}^T \check{\boldsymbol{x}}} \tag{15}$$

A selective normalized unity gradient $\hat{\nabla}_\varsigma$ is defined. It is formulated on the singlarity index using the selection vector $\boldsymbol{s}$ to restrict the movement of particles to certain coordinate directions of the gradient:

[1] *RCA562*: Robot Control Application of the Collaborative Research Center 562.

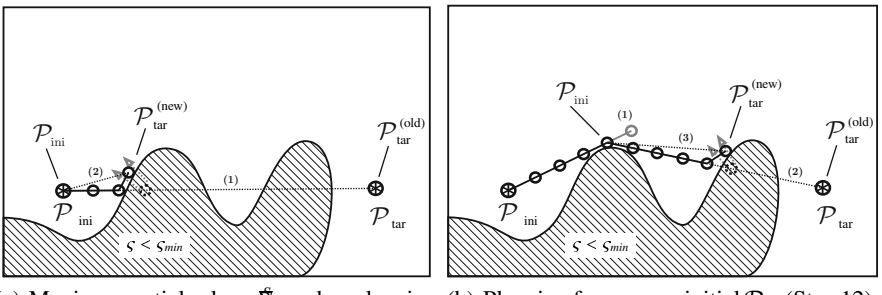
$$\vec{\nabla}_{\varsigma}(\boldsymbol{s},\boldsymbol{x}) = \frac{\mathrm{diag}[\boldsymbol{s}]\cdot\boldsymbol{N}^{-1}\cdot\nabla\varsigma(\boldsymbol{x})}{\|\mathrm{diag}[\boldsymbol{s}]\cdot\boldsymbol{N}^{-1}\cdot\nabla\varsigma(\boldsymbol{x})\|_2} \tag{16}$$

4.2 Planning Algorithm

Using the Equations (15) and (16), the base algorithm 0.1 is formulated. Its mode of operation is displayed in Fig. 3.

Algorithm 0.1 Path generation for singularity avoidance

1: initial partcle $\mathcal{P}_{\mathrm{ini}} \leftarrow$ particle at start pose $\mathcal{P}_{\mathrm{Start}}$
2: actual goal particle $\mathcal{P}_{\mathrm{tar}} \leftarrow$ particle at target pose $\mathcal{P}_{\mathrm{Ziel}}$
3: **repeat**
4: create a new particle $\mathcal{P}_n$ in the way, that a equidistant line of particles is built from $\mathcal{P}_{\mathrm{ini}}$ into the direction of $\mathcal{P}_{\mathrm{tar}}$.
5: **if** the singularity index ς of $\mathcal{P}_n$ is invalid, **then**
6: **while** the singularity index ς of $\mathcal{P}_n$ is invalid, **do**
7: move $\mathcal{P}_n$ along $\vec{\nabla}_{\varsigma}$.
8: new actual goal particle $\mathcal{P}_{\mathrm{tar}} \leftarrow$ moved particle $\mathcal{P}_n$
9: discard all particles between $\mathcal{P}_{\mathrm{tar}}$ und $\mathcal{P}_{\mathrm{ini}}$.
10: **else**
11: **if** $\mathcal{P}_n = \mathcal{P}_{\mathrm{tar}} \wedge \mathcal{P}_n \neq \mathcal{P}_{\mathrm{Ziel}}$, **then**
12: **if** a particle $\mathcal{P}_{n+1}$ according to step 4 would be valid, **then**
13: new initial particle $\mathcal{P}_{\mathrm{ini}} \leftarrow \mathcal{P}_n$
14: new actual goal particle $\mathcal{P}_{\mathrm{tar}} \leftarrow \mathcal{P}_{\mathrm{Ziel}}$
15: **until** the target pose is reached ($\mathcal{P}_{\mathrm{n}} = \mathcal{P}_{\mathrm{Ziel}}$).



(a) Moving a particle along $\vec{\nabla}_{\varsigma}$ and re-planning (step 8).

(b) Planning from a new initial $\mathcal{P}_{\mathrm{ini}}$ (Step 12).

Fig. 3. Working mode of algorithm 0.1.

Figure 3(a) shows how particles are created towards the target particle $\mathcal{P}_{\mathrm{tar}}$, until the singularity index on the dashed region is invalid (ς is undesirably low). The direction of the plannig ist annotated with (1). After that, the particle is moved with step size ϵ_v into the direction of $\vec{\nabla}_{\varsigma}$, according to the recursion on i

$$\boldsymbol{x}_{n,i} = \boldsymbol{x}_{n,i-1} + \epsilon_v \cdot \vec{\nabla}_{\varsigma}(\boldsymbol{x}_{n,i-1}) \tag{17}$$

Algorithm 0.2 Moving using a local search policy

Require: a preceeding particle $\mathcal{P}_{n-1}$ exists and the particle $\mathcal{P}_n$ at the pose $\boldsymbol{x}_{n,0}$ fulfills $\varsigma(\boldsymbol{x}_{\mathrm{n},0}) < \varsigma_{\min}$

1: **repeat**

2: move $\mathcal{P}_n$ according to Equation 17:

$$\boldsymbol{x}_{n,i} = \boldsymbol{x}_{n,i-1} + \epsilon_v \cdot \vec{\nabla}_{\varsigma}(\boldsymbol{x}_{n,i-1})$$

3: **if** the distance $\|\boldsymbol{x}_{n,0} - \boldsymbol{x}_{n,i}\|_{\mathrm{N}} < \epsilon$ is fulfilled, **then**

4: recalculate $x_{n,i}$ for $\mathcal{P}_n$ according to

$$\boldsymbol{x}_{n,i} = \epsilon \left[(1-\mu) \cdot \frac{\boldsymbol{x}_{\mathrm{tar}} - \boldsymbol{x}_{n,0}}{\|\boldsymbol{x}_{\mathrm{tar}} - \boldsymbol{x}_{n,0}\|_2} + \mu \cdot \frac{\boldsymbol{x}_{n,0} - \boldsymbol{x}_{n-1}}{\|\boldsymbol{x}_{n,0} - \boldsymbol{x}_{n-1}\|_2} \right] \tag{18}$$

5: **until** $\varsigma(\boldsymbol{x}_{\mathrm{n}})$ is valid.

until $\varsigma(\boldsymbol{x}_n) > \varsigma_{\min}$ is established. Then this particle becomes the new goal particle $\mathcal{P}_{\mathrm{tar}}$ according to step 8. The new planning direction is annotated with (2). In figure 3(b), the planning has proceeded further: Detail (1) shows the location of the particle on which the decision is made if the planning continues according to step 12 or if the planning is restarted from the new initial particle $\mathcal{P}_{\mathrm{ini}}$. Since the singularity index is valid for this pose, planning is continued towards the target particle (2), until the first moved particle becomes the new actual goal particle $\mathcal{P}_{\mathrm{tar}}$ (3).

4.2.1 Performance and Optimization

The base algorithm 0.1 solves the path planning problem. However, if the tangential components of $\vec{\nabla}\varsigma$ on the hyperplane that forms the border of invalid areas are small, particles may agglomerate. Therefore, some optimizations are proposed.

Local Search Policy

A minimum distance ϵ between the particle moved and the previous valid one is defined. If ϵ is too small, the particle is iteratively placed towards the goal particle and moved along the gradient until ϵ *and* the index ς are valid (see step 7 in algorithm 0.2). This algorithm leads to more equally distributed particles but neither reduces the number of move operations (see Eq. (17)), nor does it increase robustness against local concave areas of limited dimension. The behavior of the algorithm concerning these criteria is improved by applying step 4 of the algorithm 0.2. Using Equation (18), the particle is moved according to the weighted sum of the direction towards the goal pose and the direction of the trailing path. A further improvement is obtained when the weight of the local path direction $\mu \in [0; 1]$ is not a constant value but is increased with a rising number of moving operations.

Biologcally Inspired Optimization

In regions where the global and the local planning direction are equal and small tangential components of $\vec{\nabla}\varsigma$ of boundary hyperplanes are present, there is still potential for optimization. Motivated by the successful application of biologically inspired algorithms using pheromones by imitation of ant-like behavior [28, 29], a pheromone-based strategy can be applied. In contrast to well-known algorithms which use activating pheromomes, we propose to use inhibiting pheromones to tackle the problem mentioned above.

An area where the planning problem cannot be solved is marked with the correspondent pheromone. This area is then excluded in the future. A failed attempt can be detect when the number of iterations of step 4 of algorithm 0.2 exceeds a certain limit. If this limit is exceeded, the algorithm discards the particle, marks the pose with a pheromone particle and triggers a new planning cycle. When the singularity index is calculated, all pheromone particles are taken into account by their distance to the particle of the pose evaluated. For that purpose, the index $\varsigma = \sqrt{\lambda_{\min}}$ which is based on the optimum obtained from Equation (12) is extended by a pheromone part:

$$\tilde{\varsigma}(\boldsymbol{x}) = \varsigma(\boldsymbol{x}) + \gamma \sum_{\mathcal{P}_p} \left(1 - e^{-\frac{\|\boldsymbol{x} - \boldsymbol{x}_{\mathcal{P}_p,j}\|_{\mathrm{N}}^2}{w}} \right) \tag{19}$$

The impact of the pheromones is modelled by a Gaussian distribution function, where w adjusts the effective range of the marked poses $\boldsymbol{x}_{P_p,i}$. It may be increased progressively with the number of marked poses in order to accelerate the algorithm. Simulation results show, that the principle is succesfully applicable and in some scenarios even a single pheromone source significantly improves the performance of the path planning algorithms. Moreover, the robustness regarding widespread concave regions of the workspace is increased.

5 Conclusion

The higher the orientability of a parallel manipulator becomes, which is desirable for maximizum flexibility of the robot, the more likely singularities are located inside the workspace. Unfortunately solutions to the singularity monitoring problem based on the Jacobian typically lack unit consistency or indepency of coordinate systems. Most important is that none of them provides a physically meaningful distance measure and a globally valid limit which denotes singular regions. We proposed to tackle the problem with two methods: First a singularity analysis scheme based on Grassmann's theory which identifies all singular configurations of a mechanism was presented. However, dependent on the kinematic, the implementation of the online-monitoring might be cumbersome. This motivates the development of a physically meaningful distance measure which monitors all singularities merged in one scalar

index value, e.g. transmittable power or lowest eigenfrequency. This second class of methods overcomes the drawbacks of other singularity monitoring schemes. It was successfully demonstrated on the HEXAII parallel robot.

Additionally a path planning algorithm was developed which is capable of finding a path around singular regions. The power-based singularity index and its gradient are computed only at very few points in the workspace. Especially the amount of calculations of the computationally expensive gradient are minimized. In this way the algorithm deals efficiently with the planning problem in high dimensional spaces. The implementation on the HEXAII shows that when such a planning algorithm is available the task programmer does not have to have special knowledge about the properties of the robot involved. The robot control takes over this task during runtime and finds a suitable path that avoids singular regions on that particular robot.

Acknowledgements. The authors appreciate the support of the German Research Foundation.

References

1. Gosselin, C.M., Wang, J.G.: Singularity loci of planar parallel manipulators with revolute actuators. Robotics and Autonomous Systems 21(4), 377–398 (1997)
2. Huang, Z., Chen, L.H., Li, Y.W.: The singularity principle and property of stewart parallel manipulator. Journal of Robotic Systems 20(4), 163–176 (2003)
3. Horin, P.B., Shoham, M.: A class of parallel robots practically free of parallel singularities. Journal of Mechanical Design 130(5) (2008)
4. Choudhury, P., Ghosal, A.: Singularity and controllability analysis of parallel manipulators and closed-loop mechanisms. Mechanism and Machine Theorie 35(10), 1455–1479 (2000)
5. Ball, R.S.: A Treatise on the Theory of Screws. Cambridge University Press, Cambridge (1900)
6. Hunt, K.H.: Don't cross-thread the screw. In: Proceedings of a Symposium Commemorating the Legacy, Works, and Life of Sir Robert Stawell Ball upon the 100th Anniversary of A Treatise on the Theory of Screws, pp. 1–36 (2000)
7. Hunt, K.H.: Kinematic Geometry of Mechanisms. Oxford University Press, New York (1978)
8. Poinsot, L.: Sur la composition des moments et la composition des aires. Éc Polyt Paris 6, 182–205 (1806)
9. Bier, C.: Geometrische und physikalische Analyse von Singularitüten bei Parallelstrukturen. Ph.D. thesis, Fakultät für Maschinenbau, TU Braunschweig, Vulkan-Verlag (2006)
10. Sciavicco, L., Siciliano, B.: Modeling and Control of Robot Manipulators. Springer, London (2000)
11. Tsai, L.-W.: Robot Analysis - The Mechanics of Serial and Parallel Manipulators. John Wiley and Sons Inc., New York (1999)
12. Gosselin, C., Angeles, J.: Singularity analysis of closed-loop kinematic chains. IEEE Transactions on Robotics and Automation 6(3), 281–290 (1990)
13. Simaan, N., Shoham, M.: Singularity analysis of class of composite serial in-parallel robots. IEEE Transactions on Robotics and Automation 17(3), 301–311 (2001)

14. Kozak, K., Voglewede, P.A., Ebert-Uphoff, I., Singhose, W.: Concept paper: On the significance of the lowest linearized natural frequency of a parallel manipulator as a performance measure for concurrent design. In: Proceedings of the Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators, Quebec City (Kanada), pp. 112–118 (2002)
15. Yoshikawa, T.: Manipulability of robotic mechanisms. The International Journal of Robotics Research 4(2), 3–9 (1985)
16. Lee, J., Duffy, J.: The optimum quality index for some spatial in-parallel devices. In: Proceedings of Conference on Recent Advances in Robotics, Miami, USA, pp. 1–19 (1997)
17. Angeles, J., López-Cajún, C.S.: Kinematic isotropy and conditioning index of serial robotic manipulators. The International Journal of Robotics Research 11(6), 560–571 (1992)
18. Gosselin, C.: Stiffness mapping for parallel manipulators. Transactions on Robotics and Automation 6(3), 377–382 (1990)
19. Doty, K.L., Melchiorri, C., Schwartz, E.M., Bonivento, C.: Robot manipulability. IEEE Transactions on Robotics and Automation 11(3), 462–468 (1995)
20. Duffy, J.: The fallacy of modern hybrid control theory that is based on orthogonal complements of twist and wrench spaces. Journal of Robotic Systems 7(2), 139–144 (1990)
21. Lipkin, H., Duffy, J.: Hybrid twist and wrench control for a robot manipulator. Journal of Mechanisms, Transmissions, and Automation in Design 110, 138–144 (1988)
22. Zanganeh, K.E., Angeles, J.: Kinematic isotropy and optimum design of parallel manipulators. The International Journal of Robotics Research 16(2), 185–197 (1997)
23. Hesselbach, J., Bier, C., Campos, A., Löwe, H.: Direct kinematic singularity detection of a hexa parallel robot. In: Proceedings of IEEE International Conference of Intelligent Robots and Systems, Edmonton (Kanada), pp. 3249–3254 (2005)
24. Bier, C., Queiroz, E., Campos, A., Maass, J., Guenther, R.: Direct singularity avoidance strategy for the hexa parallel robot. In: 18th International Congress of Mechanical Engineering (2005)
25. Maaß, J.: Ein Beitrag zur Steuerungstechnik für parallelkinematische Roboter in der Montage. Ph.D. thesis, Fakultät für Maschinenbau, TU Braunschweig, Vulkan Verlag, 2009 (in german)
26. Maaß, J., Kolbus, M., Bier, C., Wobbe, F., Schuhmacher, W., Raatz, A., Hesselbach, J.: Advances in motion control for high-performance parallel robots. In: Robotic Systems for Handling and Automation, pp. 285–296 (2008)
27. Dietrich, F., Maaß, J., Raatz, A., Hesselbach, J.: RCA562: Control Architecture for Parallel Kinematic Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 315–331. Springer, Heidelberg (2010)
28. Brueckner, S.: Return from the Ant. Ph.D. thesis, Humboldt-Universitat Berlin (2000)
29. Hsiao, Y., Chuang, C., Chien, C.: Ant colony optimization for best path planning. In: IEEE International Symposium on Communications and Information Technology, ISCIT 2004, vol. 1 (2004)

Calibration of Parallel Kinematic Structures – Overview, Classification and Comparison

Philipp Last, Annika Raatz, and Jürgen Hesselbach

Abstract. Kinematic calibration has been shown to be an efficient method to improve the absolute accuracy of industrial robots. Originally developed for serial manipulators a number of methods have been adopted to the class of parallel robots. The most common calibration techniques rely on external measurement systems, which are expensive, time consuming and require highly skilled operators. Newest developments focus on autonomous self-calibration techniques which allow for easy calibration if required, without particular knowledge about robot calibration. Some promising approaches exist, especially for parallel robots. This paper classifies the different robot calibration schemes, presents some innovative calibration ideas and compares them to existing strategies. Prospects as well as problems of the solutions are discussed.

1 Introduction

Instead of teach-in programming offline programming is a means to generate robot programs without occupying the robot itself during the program generation step. It is thus supposed to be more economical advantageous than teach-in programming. However, although a lot of research work has been dedicated to this topic and it can be said to be state of the art, robot offline programming is not yet often applied [1]. The main reason is a limited absolute accuracy of most robots which is

Philipp Last
Siemens AG, Dept. E D MV C R&D 4, Nonnendammallee 104, 13629 Berlin, Germany
e-mail: philipp.last@siemens.com

Annika Raatz · Jürgen Hesselbach
Technische Universität Braunschweig, Institute of Production Automation and Machine Tools, Langer Kamp 19b, 38106 Braunschweig, Germany
e-mail: {a.raatz,j.hesselbach}@tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 93–106.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

due to the fact that robot control is usually model based. That means, if a pose given in cartesian coordinates shall be reached by the robot, the controller fist computes the corresponding motor coordinates, by an internal model and subsequently starts moving the actuators towards these coordinates. Obviously, when the model does not exactly reflect the real robot geometry, the computation results in wrong motor coordinates and the robot reaches a position which does not exactly correspond to the desired one. If the deviation between the desired pose and the reached pose is small, a robot is said to feature a high absolute accuracy; otherwise the robot's absolute accuracy is low.

Because poses in an offline generated robot program are always defined in cartesian coordinates, a sufficient absolute accuracy is required in order to precisely reach the desired poses. Unfortunately deviations between the nominal geometry of a robot manipulator and the real geometry are fairly common and not negliable. They mainly result from manufacturing and assembly tolerances. Thus, absolute accuracy is often limited and hence offline generated programs cannot be sufficiently precisely executed.

A method to overcome this drawback is termed robot calibration [2]. Different approaches exist where model based calibration as explained in the following section is the most common way. Usually industrial robots are precalibrated before delivery in order to guarantee a certain absolute accuracy. Additionally, due to mechanical wear, crash oder due to errornous assembly it can be required to recalibrate a robot during life-time. Methods which are suited for calibration and especially for recalibration are presented, compared and discussed within this paper.

2 Model Based Calibration - General Approach

The most common way for kinematic robot calibration is termed model-based calibration. The underlying idea is to estimate the parameters of the kinematic model in a way that the model respresents the real mechanical structure as closesly as possible and to correct the model in the controller afterwards. In order to do so the essential point is to provide measurements by which it is possible to formulate a residual

$$\mathbf{r}_i = \tilde{\mathbf{b}}_i - \mathbf{b}_i(\mathbf{k}) \tag{1}$$

with $\tilde{\mathbf{b}}_i$ some redundantly measurement information, $\mathbf{b}_i(\mathbf{k})$ a vector with corresponding information provided by the kinematic model and $\mathbf{k}$ the parameter vector which is supposed to be identified. If such a residual can be obtained at n different measurement configurations then it is possible to stack all the information in a residual vector $\mathbf{r} = [\mathbf{r}_1^T, \mathbf{r}_2^T, \ldots, \mathbf{r}_n^T]^T$. Once this vector is available, it is the goal to estimate the parameters in such a way that

$$\mathbf{r} = \mathbf{0} \tag{2}$$

Due to measurement noise and model simplifications this goal is, however, of theoretical nature and will never be exactly reached in reality. Instead, one aims to minimize a cost function $F = \mathbf{r}^T\mathbf{r}$, which if $\mathbf{r} = \mathbf{0}$ would be fulfilled equals zero as

well and otherwise is bigger than zero. The minimization of F can be attained by any optimization methods in principle. Usually, due to the special so-called least squares form of the function F, least square algorithms such as the Levenberg-Marquardt approach are applied. Minimization of F yields a parameter vector $\mathbf{k}^{kalib}$ which is then used to replace the original parameters that were used before calibration within the robot-controller.

Considering the aforementioned remarks, four essential steps can be identified which are existent in each model-based calibration approach. These are [2]:

1. **Modelling**
   In the modelling phase a kinematic model is set up which includes a number of geometric parameters that are supposed to be identified by calibration
2. **Measurement**
   The measurement step provides the redundant information required for calibration
3. **Parameter Identification**
   By means of feasible mathematical methods the model parameters are identified in a way so that model and measurements correspond to each other in a best possible way
4. **Parameter Correction**
   Within the parameter correction step the identified parameters are transferred to the robot controller

Originally developed for serial robots, the general model based robot calibration approach as explained before can equivalently be applied to the class of parallel robots. This class of robots is characterized by a specific kinematic layout. Compared to a conventional industrial robot, a parallel robot based on closed kinematic chains consists of at least two driven guiding chains to guide a common working platform with a gripper or a moving tool attached to it [3]. The possibility to mount all drives in the frame or near to the frame results in low moved masses allowing high operating speeds and accelerations. Due to the fact that the end-effector is supported by several guiding chains, further advantages are a high structural stiffness and a modular design. During the past years parallel robots proved to be an efficient and suitable supplement to serial robots based on open kinematic chains.

3 Classification of Different Calibration Techniques

A huge number of model based calibration methods already exist which follow the general scheme described in the preceding section. Differences between these techniques can be found with respect to various aspects at different stages. The most obvious and most important differences, however, exist in the measurement phase. Based on this appraisal a general classification can be defined for the different calibration strategies which includes the two seperation criteria: 1. degree of automation and 2. data-acquisition method, both briefly explained in what follows.

- **Degree of Automation**
  In regard to the degree of automation autonomous and non-autonomous calibration techniques are distinguished. A calibration method is understood to work autonomously only if all steps of the overall procedure can be completely automated and absolutely no user interaction is required during calibration. If any effort is needed for preparation, for accomplishment or data-transfer during calibration then the corresponding technique is defined to be non-autonomous. It should be noted that the non-autonomous methods, although combined in one group, may drastically vary in the amount of required manual support.
- **Data-Acquisition Method**
  Two fundamental data-acquisition methods may be used for robot calibration. The first one uses additional sensors (internal or external) which are not required for operating the robot but are necessary to achieve redundant information. The second method relies on kinematic constraints which are introduced in the system without raising the number of sensors. In this case, due to constraints the actuator measurement systems which are already part of the robot system deliver enough information for robot calibration.

In combination of all possible classification attributes there are four principle types of calibration techniques, namely type A, type B, type C and type D (see Fig. 1). Specific calibration methods of each type are presented in the next section.

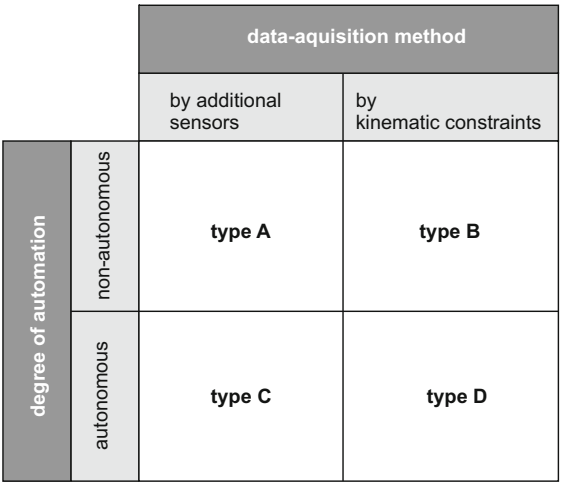


Fig. 1. Classification of robot calibration techniques

4 Presentation of Selected Calibration Methods

To clarify the aforementioned classification, different exemplarily chosen calibration techniques are concretely described in this paragraph. This list contains commonly known techniques already in industrial use as well as rather new calibration

schemes featuring promising characteristica, but need to be further developed in order to fit industrial needs. A focus is on methods suited for parallel robots.

4.1 Type A Methods

Type A calibration techniques rely on additional external measurement equipment which is characterized by the fact that it is not — at least not permanently — part of the robot system. Usually, for calibration some preparation is necessary in order to set up the additional measurement system. Moreover, some manual work might be required for data transfer or during accomplishment of the measurement phase within the calibration process. This explains why these calibration techniques are non-autonomous.

A typical example of a type A calibration method is to use a lasertracker system (see Fig. 2) which is capable of measuring an absolute position of a reflector target in space. For this, a laser-beam is sent to the reflector target, which can be mounted anywhere on the robot structure in principal. Then through a combination of angle and interferometric distance measurement the 3D-point coordinates of the reflector target are evaluated. If the reflector is attached to the end-effector of a robot, then by means of this special sensor the end-effector position of a robot can be evaluated and compared to corresponding information obtained through the kinematic model in a residual.

Following the same principal alternative measurement systems, such as theodolites, camera-systems or cable-based measurement systems can be applied. Moreover, similar approaches based on different sensors exist in which not the endeffector position but the complete pose (position and orientation of the endeffector) or the distance to a fixed point in space is measured and compared to corresponding model information.

Type A calibration techniques are very general and can be applied not only to parallel kinematic structures but also to serial ones. Moreover, they are well established in industry and can thus be seen as the standard type calibration technique.

4.2 Type B Methods

Redundancy in type B methods is obtained by lowering the system's degree of freedom (DOF) through some mechanical device (no sensor) while still considering all motor-readings. Because at least for attaching the apparatus to the structure involves some manual work type B methods are also non-autonomous.

DOF-reduction can, e.g., be achieved by restraining the endeffector's motion. A way, that is often traced and which has been described e.g. in [4] is based on contour following. That means, a body of known contour, e.g. a cylinder or a plate is placed within the workspace of the robot. Then the robot-end-effector is sequentially guided to a huge number of points on the contour. For each point the end-effector coordinates can be computed through the kinematic model and then the parameters of the model are identified in a way such that the computed contour and the known one of the contour-body correspond to each other as closesly as possible. The described

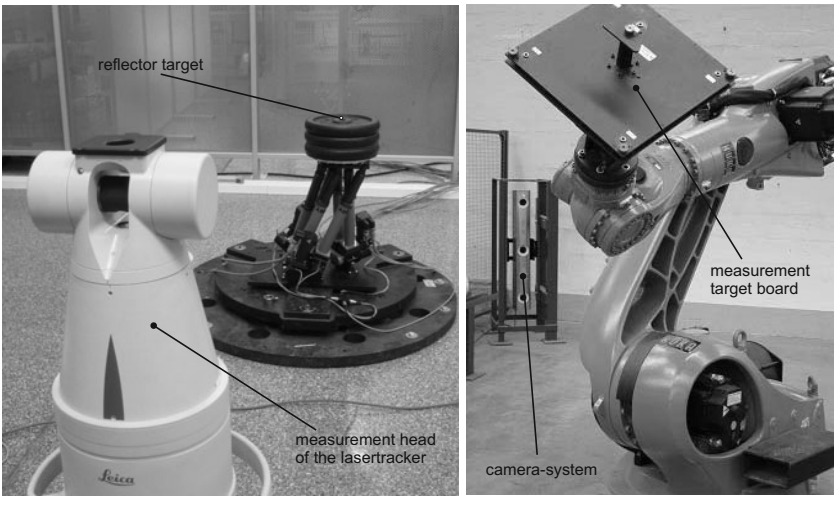


Fig. 2. External measurement systems used fpr robot calibration: Lasertracker (left) and camera system (right)

approach can be conveniently applied if force-control is implemented, otherwise contour touching is a process characterized by a huge amout of manual work and calibration will take a lot of time. Robot calibration by contour following is again very general and can be applied to serial and parallel robots.

For parallel robots only a different approach exists which also belongs to type B methods. The basic idea is to use clamping devices forcing passive joints of a parallel kinematic structure to remain in a specified position. In that case the structure can be controlled by a reduced number of actuators while the position-sensors of the remaining motors deliver redundant information which is used for calibration. A drawback of this approach is, that the workspace of the system with reduced DOF is usually very small resulting in measurements which are obtained in a comparatively small area of the original robot workspace only. It is often claimed that due to this the described calibration approach is less accurate than others. Another disadvantage with passive-joint-clamping is, that it must be garantueed, that motors may move passively. This is in practice hard to achieve [3] - a possible method to solve this problem, which was originally intended for another purpose, is described in [5].

In order to overcome the problem with measurements obtained only in a limited workspace area a new concept which is similar to the one described before has been proposed in [6]. Instead of using passive clamping devices the passive joints in a parallel kinematic manipulator are replaced by so-called adaptronic joints. These are passive joints equipped with a piezo-actuator which allows to adjust the joint play. By this means it is possible to support high precision tasks such as assembly operations by reducing joint play. On the other hand if high dynamics are required, e.g. in handling tasks, then joint play may be increased so that friction gets minimal. The development and a first prototype of this kind of joint are described in [7]. In

an extreme case joint play is reduced in a way so that friction is extremely high and the joint motion is completely blocked. This blocking feature can be used instead of a clamping device to force a passive joint to remain in a certain position, thereby reducing the structure's DOF. The calibration is then executed in the same way as before but with the advantage that the joint can be blocked in arbitrary positions allowing to obtain measurement information almost everywhere in the robot's workspace.

Exemplarily a calibration based on the adaptronic joints has been simulated for the FIVEBAR-robot [6]. A typical result of the corresponding investigations is shown in Fig. 3. It depicts the absolute positioning error before and after calibration at a number of control configuration spread over the whole workspace. Because the positioning error is drastically decreased after calibration we conclude that the method is successful.

Type B methods are in general relatively inexpensive because no complex measurement-systems are required. Another advantage lies in the fact that the only sensor information to be considered is that of the actuator measurement systems so that the identification algorithms can be directly implemented on the robot controller.

4.3 Type C Methods

As well as the calibration techniques described in section 4.1, the methods belonging to type C rely on additional measurement devices. However, in contrast to the methods explained above, the sensors used in type C-techniques are internal sensors. That means, they are permanently mounted on the robot so that it is possible to read and process their sensor information within the control system of the robot. This enables to completely automate the calibration process.

While equipping serial robots with additional internal sensors is complex it is very convenient to insert the sensors to the passive joints of a parallel robot. Not only a complete automation of the calibration process is possible but a further advantage lies in the fact that measurements with equal accuracy can be obtained within the

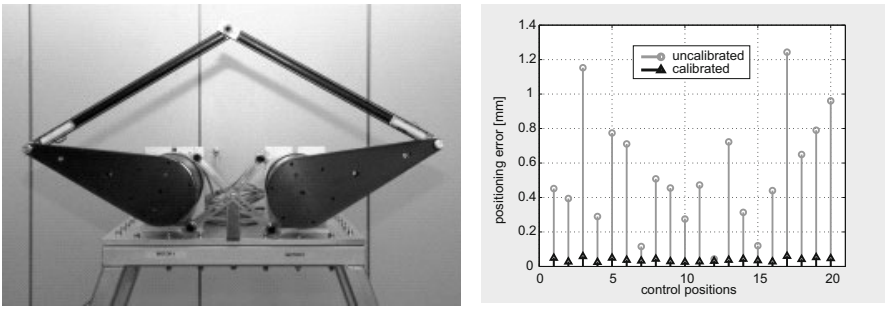


Fig. 3. FIVEBAR parallel robot [8] and corresponding calibration result achieved by calibration based on adaptronic joints.

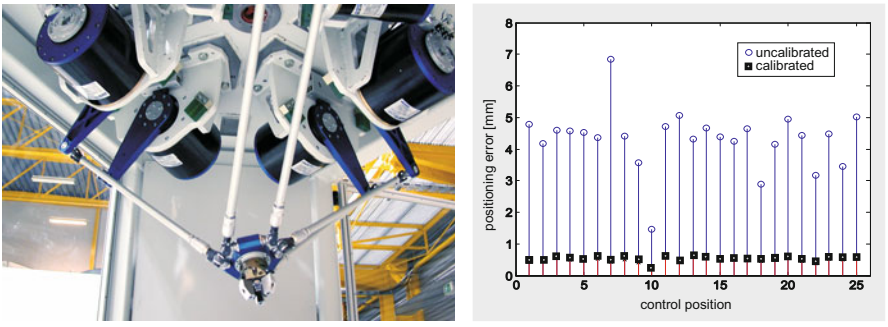


Fig. 4. HEXA-parallel robot [8] and corresponding calibration simulation result achieved by joint integrated microsensors

whole workspace. Obviously, using internal sensors makes a robot more expensive. However, it should be considered that the sensors may facilitate different control functionalities, e.g. workspace monitoring [9], thus making passive joint sensing even more interesting. Although the use of joint-integrated sensors is very promising (especially for calibration purposes) application is limited to research institutions so far. This is because adding passive joint-sensors to a structure is not an easy task, particularly due to the challenging sensor requirements [10, 11]: Joint integration forces sensors to be small in dimension while high resolution is required. Furthermore, the rough environmental conditions inside the joints such as lubrication and vibration have to be kept in mind. Finally the sensor mass has to be as small as possible in order to limit dynamic feedback effects. Additional integration of sensors into the passive joints must be considered during the design process because adding passive joint sensors to an existing structure is even more complex. A feasible combined sensor-joint-design is presented in [10, 12].

Fig. 4 shows a simulation-result achieved by simulating a calibration of the HEXA-robot based on measurements achieved by joint-integrated micro-sensors [13, 14]. The positioning error before and after calibration is compared at different randomly chosen control positions within the robot's workspace. Obviously the accuracy can be significantly improved by this method, so that the method can be said to be successful.

Besides adding joint sensors to the passive joints of a manipulator a further approach for robot calibration, also classified as type C, consists in the use of additional sensor-equipped kinematic chains between base and endeffector. These can be actuated or unactuated chains. In every case the additional kinematic chains must not limit the DOF; otherwise they cannot be permanently part of the robot structure. The design and the use of an unactuated kinematic chain for robot calibration are described in [15]. Robot calibration by means of an additional actuated kinematic chain is investigated in [16].

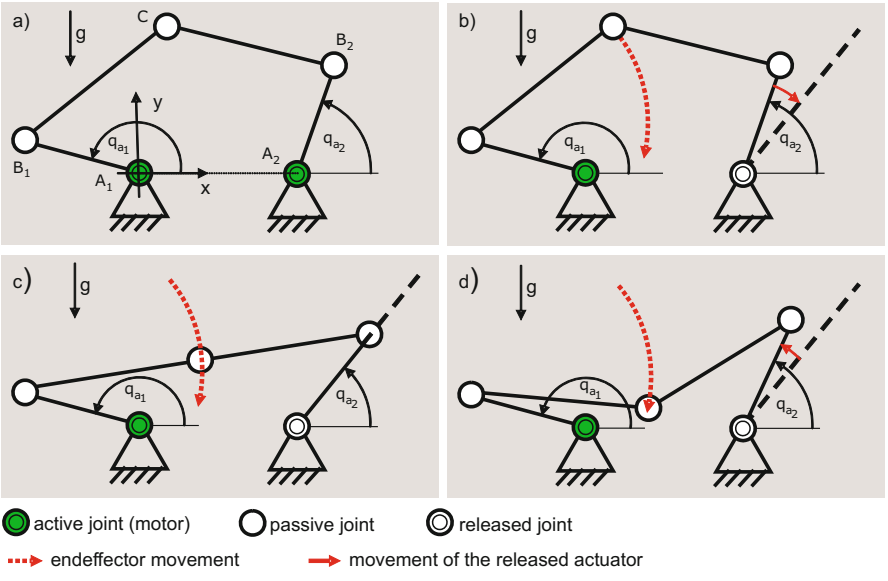


Fig. 5. Change of direction of movement during singularity passing

4.4 Type D Methods

Type D calibration approaches are rare. Up to the authors' knowlege there is only one technique which is completely automatable and relies on kinematic constraints. This method is called the singularity calibration technique and has been introduced in [17]. The fundamental idea of this technique is to achieve redundancy by special knowledge about so-called singular configurations of type 2 which need to be passed in order to identify the kinematic parameters of a parallel kinematic manipulator. These are robot configurations in which several solutions of the direct kinematic problem coincide. Since those kind of singularities do not occur in a serial robot's workspace, the singularity based method just holds for parallel robots. Usually approaching type 2 singularities is avoided as the robot gets uncontrollable in those configurations, thus comprising a risk of structure or even robot-user damaging. However, in [18] an approach has been developed to savely pass singularities of type 2 with the intention to enlarge a robot's workspace. The basic idea is to temporarily underactuate the robot system by releasing one of the actuators and to use some additional driving force to guide the parallel kinematic structure through the singularity of type 2. In Fig. 5 the process is visualized by means of the FIVE-BAR-robot, already known from Fig.3. In a) the structure is shown in a non-singular position. By releasing one actuator b) the structure gets underactuated. Assuming that gravity acts on the system, the endeffector point C starts moving. At a certain time the manipulator passes the singularity of type 2 c) and a little bit later it reaches a non-singular configuration d) in which the released actuator can be activated again. Instead of exploiting gravity as the driving force which has been also done in [19],

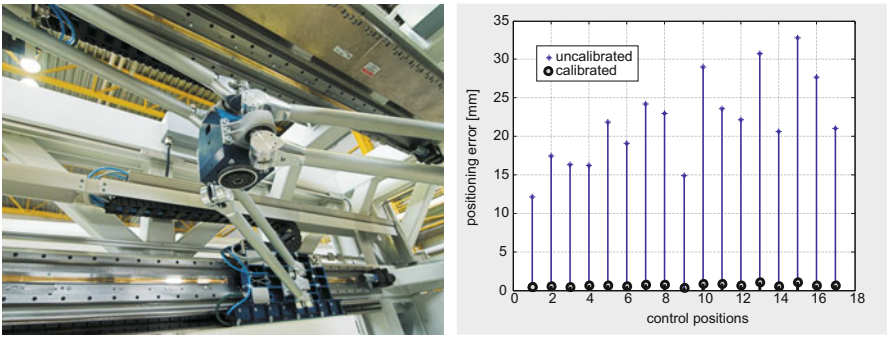


Fig. 6. TRIGLIDE parallel robot [8] and corresponding simulation result achieved by singularity based calibration

structure inertia may be used to pass the singularity as described in [18]. Following this approach, it can be found that the released actuator changes its direction of movement exactly in the point of the singularity of type 2 as far as it can be guaranteed that all other motors (motor 1 in the example) hold their position. Hence, observing the actuator movement of the released actuator it is possible to identify the singular configuration at the time where the actuator velocity is zero. The corresponding actuator coordinate can be saved. Furthermore, since particular geometric conditions need to be fulfilled at a singular configuration of type 2, it is possible to compute the actuator coordinate from the kinematic model including the kinematic parameters. Comparing both information allows to formulate a residual function of the form given by equation (1) and hence calibration is possible.

The proposed technique could be confirmed in different simulation studies for various parallel structures [20, 21]. A condition for the application is that the parallel kinematic mechanism to be calibrated is designed in a way, so that singularities of type 2 may be passed. Despite this requirement the method is general and has the advantage that it can be completely automated without the necessity to use any special calibration equipment.

A typical simulation-result achieved by investigating singularity calibration of the TRIGLIDE-structure is depicted in Fig. 6. Again the positioning error after calibration is compared to the error before calibration and it turns out that singularity based calibration is a viable means to enhance the absolute accuracy.

5 Comparison

As the remarks in sections 4.1 — 4.4 show, a huge number of different calibration techniques exist, each one characterized by specific technical and economical advantages and drawbacks. From the technical point of view the most important goal is the accuracy enhancement to be achieved by a calibration method. However, it is hard to specify the accuracy improvement for a group of calibration methods

and even for a certain calibration approach in general, because it not only depends upon the applied equipment but also strongly depends on the mechanical structure to be calibrated. Besides accuracy one of the most important criteria which affects the success of a calibration-scheme is cost [22]. Two different cost-aspects can be distinguished. These are equipment costs as well as labour costs. Labour costs are directly related to the degree of automation. Obviously, if a calibration technique can be completely automated then the labour costs are negliable. For non-autonomous calibration methods on the other hand there are labour costs. As stated above, these may vary, because the amount of manual work differs drastically for the different non-autonomous calibration methods. The equipment costs depend upon the device related effort. Depending on the calibration strategy the expenses related with the required calibration equipment acquisition are not negliable. E.g., the acquisition cost for a Lasertracker is about 150.000 €. Especially for small and medium sized companies with a small number of robot units which need to be recalibrated from while to while this price is not affordable. In Fig. 7 the different types of calibration are qualitatively arranged within a diagram that allows to rank the degree of automation and at the same time the device related effort. It thus allows for a rough comparison of the different types of calibration.

Obviously for type A and type C calibration methods the device related effort and hence the equipment costs are comparetively high. This is because for both types complex sensor systems are used. The passive mechanical devices which are usually applied in order to achieve redundancy in type B calibration methods result in a device related error which is smaller. Finally, for the presented singularity based calibration technique classified as type D method, the device related effort is minimal. In this case no special calibration equipment is required, since redundancy is obtained by singularities which are structure inherent constraints.

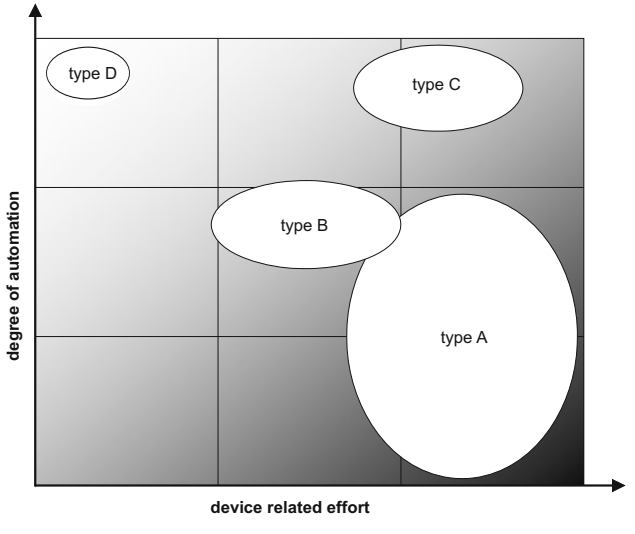


Fig. 7. Qualitative classification of existing calibration techniques of different type

With respect to the degree of automation type C and type D methods are most appropriate, because as explained, they are completely automatable. The degree of automation for type A as well as for type B methods is lower. In tendency type B methods are slightly stronger automated than type A methods. This is because for type B strategies all the sensor information is obtained by the motor measurement systems so that the information can be directly processed in the robot controller and at least for data transfer no manual effort is required.

It can be concluded, that singularity based calibration as the only member of type D calibration techniques has the best characteristics and shows an edge over the alternative methods. Hence, from an economical point of view it is the preferred technique and should be applied whenever technical performance is sufficiently good.

6 Conclusion

A number of interesting calibration methods exist, all of them capable of improving the absolute accuracy of robotic systems. In an overview several of these techniques have been briefly presented, classified and qualitatively compared to each other. It turns out that escpecially for parallel robots very promising methods exist, which feature several advantages compared to those technqiues (mainly those of type A) which are already in industrial use. E.g., they allow for easier and faster calibration while being comparetively inexpensive. These positive characteristics make the new calibration strategies interesting in general and in particular for small and medium sized companies, which aim to recalibrate their systems during lifetime but do not own special and expensive calibration equipment and do not employ high skilled operators who are able to operate the complex measurement systems which are usually required in todays calibration methods.

In order to take advantage from these new techniques it is, however, required that robot systems provide the necessary functionalities. E.g. parameter identification algorithms and parameter correction methods should be implemented in the robot controller in order to simplify the calibration process for the user. Taking advantage from this demand is a chance for robot and/or robot control manufacturers, so that both, the robot-users as well as the robot manufacturers may profit from the proposed calibration techniques.

Acknowledgements. The research work reported here was supported by the German Research Foundation (DFG) within the scope of the Collaborative Research Center SFB 562.

References

1. Wiest, U.: Kinematische Kalibrierung von Industrierobotern. In: Berichte aus der Automatisierungstechnik, Shaker Verlag, Aachen (2001)
2. Mooring, B.W., Roth, Z.S., Driels, M.R.: Fundamentals of manipulator calibration. A Wiley-Interscience publication, New York (1991) ISBN 978-0471508649

3. Merlet, J.P.: Parallel Robots, 2nd edn., vol. 128. Springer, Dordrecht (2006) ISBN 978-1402041327
4. Legnani, G., Adamini, R., Jatta, F.: Calibration of a scara robot by force-controlled contour tracking of an object of known geometry. In: Proceedings of the 32nd International Symposium on Robotics (2001)
5. Maaß, J., Kolbus, M., Budde, C., Hesselbach, J., Schumacher, W.: Control strategies for enlarging a spatial parallel robot's workspace by change of configuration. In: Proceedings of the 5th Chemnitz Parallel Kinematics Seminar, Chemnitz, Germany, pp. 515–530 (2006)
6. Last, P., Raatz, A., Hesselbach, J., Pavlović, N., Keimer, R.: Parallel robot calibration utilizing adaptronic joints. In: ASME 32nd Annual Mechanisms and Robotics Conference, New York, USA (2008)
7. Pavlović, N., Otremba, R., Inkermann, D., Franke, H.J., Vietor, T.: Passive and Adaptive Joints for Parallel Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 429–444. Springer, Heidelberg (2010)
8. Schütz, D., Budde, C., Raatz, A., Hesselbach, J.: Parallel Kinematic Structures of the SFB 562. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 109–124. Springer, Heidelberg (2010)
9. Hesselbach, J., Bier, C., Pietsch, I., Plitea, N., Büttgenbach, S., Wogersien, A., Güttler, J.: Passive joint-sensor applications for parallel robots. Mechatronics 15(1), 43–65 (2005)
10. Büttgenbach, S., Güttler, J., Last, P., Bier, C., Otremba, R.: Development of angular joint-sensors and application to parallel robots. In: Last, P., Budde, C., Wahl, F.M. (eds.) Proceedings of the 2nd International Colloquium of the Collaborative Research Center 562: Robotic Systems for Handling and Assembly. Fortschritte in der Robotik, vol. 9, pp. 237–251. Shaker Verlag, Braunschweig (2005)
11. Güttler, J., Last, P., Otremba, R., Büttgenbach, S.: A novel angular joint-sensor using a fluxgate magnetometer. In: Proceedings of the 4th IEEE International Conference on Sensors, Irvine, USA, pp. 53–56 (2005)
12. Kirchhoff, M.R., Güttler, J., Wogersien, A., Pavlović, N., Otremba, R., Franke, H.J., Büttgenbach, S.: Design and Implementation of Adaptronic Robot Components. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 413–427. Springer, Heidelberg (2010)
13. Hesselbach, J., Bier, C., Budde, C., Last, P., Maaß, J., Bruhn, M.: Parallel robot specific control functionalities. In: Last, P., Budde, C., Wahl, F.M. (eds.) Proceedings of the 2nd International Colloquium of the Collaborative Research Center 562: Robotic systems for handling and assembly. Fortschritte in der Robotik, vol. 9, pp. 93–108. Shaker Verlag, Braunschweig (2005) ISBN 3-8322-3866-2
14. Last, P., Budde, C., Hesselbach, J.: Self-calibration of the HEXA-parallel-structure. In: Proceedings of the IEEE Conference on Automation Science and Engineering, Edmonton, Canada, pp. 393–398 (2005)
15. Rauf, A., Pervez, A., Ryu, J.: Experimental results on kinematic calibration of parallel manipulators using a partial pose measurement device. IEEE Transactions on Robotics 22(2), 379–384 (2006)
16. Zhang, Y., Cong, S., Li, Z., Jiang, S.: Auto-calibration of a redundant parallel manipulator based on the projected tracking error. Archive of Applied Mechanics 77(10), 697–706 (2007)
17. Last, P., Hesselbach, J.: A new calibration strategy for a class of parallel mechanisms. In: Lennarčič, J., Roth, B. (eds.) Advances in Robot Kinematics, pp. 331–338. Springer, Dordrecht (2006) ISBN 978-1-4020-4940-8

18. Helm, M.B.: Durchschlagende Mechanismen für Parallelroboter. In: Schriftenreihe des Instituts für Werkzeugmaschinen und Fertigungstechnik der TU Braunschweig, Shaker Verlag, Essen (2003)
19. Budde, C.: Wechsel der Konfiguration zur Arbeitsraumvergrößerung bei Parallelrobotern. In: Schriftenreihe des Instituts für Werkzeugmaschinen und Fertigungstechnik der TU Braunschweig. Vulkan Verlag, Essen (2009) ISBN 978-3802787478
20. Last, P., Budde, C., Krefft, M., Hesselbach, J.: Parallel robot self-calibration without additional sensors or constraint devices. In: Proceedings of the ISR/Robotik Joint Conference, VDI, Munich, Germany, vol. 1956, pp. 269–270 (2006)
21. Last, P., Schütz, D., Raatz, A., Hesselbach, J.: Singularity based calibration of 3-dof fully parallel planar manipulators. In: Proceedings of the 12th IFToMM World Congress in Mechanism and Machine Science, Besançon, France (2007)
22. Hidalgo, F., Brunn, P.: Robot metrology and calibration systems: A market review. Industrial Robot: An International Journal 25(1), 42–47 (1998)

Part II
Implemented Systems

Parallel Kinematic Structures of the SFB 562

Daniel Schütz, Christoph Budde, Annika Raatz, and Jürgen Hesselbach

Abstract. The objective of the Collaborative Research Center (SFB) 562 "Robot Systems for Handling and Assembly - Highly Dynamic Parallel Structures with Adaptronic Components" is the research of fundamental methods and components for the development of parallel robot systems with regard to high operating speeds, accelerations and accuracy. Eight institutes in the field of mechanical and electrical engineering as well as computer science at the Technische Universität Braunschweig and the German Aerospace Center (DLR) are cooperating to investigate theoretical results as well as the exploration of new concepts on real robots. These robots which have been developed since the formation of the SFB 562 ten years ago are the topic of this paper.

1 Introduction

The stiff and lightweight structures of parallel robots provide highly dynamic movement combined with high precision. On the other side, this comes with certain drawbacks such as immobility, singularities within the workspace and a small ratio of workspace to installation space. The challenge in further development of parallel robots is eliminating or bypassing these weaknesses of parallel structures and to extend their strengths. Therefore the members of the SFB 562 investigate new approaches in the fields of kinematics, design and control. To give a general overview

Daniel Schütz · Annika Raatz · Jürgen Hesselbach
Technische Universität Braunschweig, Institute of Production Automation and Machine Tools, Langer Kamp 19b, 38106 Braunschweig, Germany
e-mail: {d.schuetz,a.raatz,j.hesselbach}@tu-bs.de

Christoph Budde
ABB Corporate Research Center Germany, Wallstadter Straße 59, 68526 Ladenburg, Germany
e-mail: christoph.budde@de.abb.com

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 109–124.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

of each parallel robot developed by the SFB 562, a representation of the kinematic structure, performance data and the motivation for building the specific robot are the aim of this contribution.

2 The FIVEBAR-Robot

The aim of the SFB 562 is to improve parallel robots not only with new optimized structures providing high dynamics and good accuracy but also with new robot control approaches. Therefore a prototype of the classical FIVEBAR-robot with a RRRRR kinematic structure (Fig. 1) has been built with five rotational joints (R), lightweight materials and additional adaptronic components including actuators and sensors in the load path.

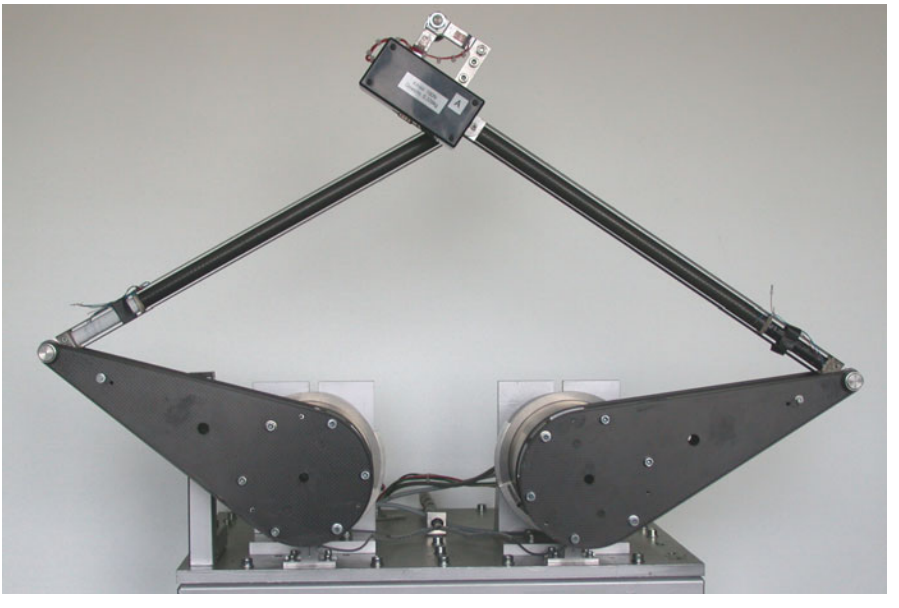


Fig. 1. FIVEBAR-robot of the SFB 562.

2.1 Design Features

The RRRRR kinematic structure of the FIVEBAR-robot has five links connected by five revolute joints in a closed kinematic chain. Two rotational drives are mounted to the frame, which represent the first and the fifth revolute joint and constitute the active joints in the structure, respectively. The other three joints are passive so that the overall degree of freedom (DOF) of the structure is two. The prototype of the FIVEBAR-structure is installed upright in such a way that forces resulting from gravity only act within the plane of motion.

The geometric parameters as shown in Tab. 1 are selected in order to obtain an improved dynamic behavior and to provide low masses. This allows for increased acceleration, as the variables have been optimized to take advantage of the maximum torque of the rotational drives.

Table 1. Geometric parameters of the FIVEBAR-structure.

Parameter description	Symbol	Value
Distance between the rotational drives	a	300 mm
Crank length	b_i	300 mm
Crank diameter on the side of the drives	d_i	149 mm
Rod length	c_i	500 mm

2.2 Performance Charateristics

The achievable dynamics and other specifications of the FIVEBAR-robot are listed in Tab. 2.

Table 2. Performance data of the FIVEBAR-robot.

Description	Data
Degree of freedom	2
Workspace x,z	400 x 600 mm^2
Max. payload	1 kg
Max. velocity	4 m/s
Max. acceleration	1 g

2.3 Fields of Research

The main objective is to enhance the performance of the FIVEBAR-robot such as the precision of the trajectory. One method to achieve this aim is to reduce vibrations resulting from high dynamics and to reduce clearance in the joints. Furthermore, the cranks are made from carbon fiber reinforced plastics to reduce the moving masses. The two rods are constructed from carbon fiber with integrated piezo stacks, which are utilized for new control approaches [1, 2, 3, 4]. The mechanical impedance of the structure can be manipulated with these adaptronic rods as well as with newly developed adaptronic joints. Thus, the velocity of the FIVEBAR-robot is improved in the vicinity of contact points [5, 6, 7, 8]. Hence, it is possible to increase the structural damping by controlling the friction in the joints. While the robot is in motion, the control provides high stiffness needed for good path accuracy, whereas

it guarantees low stiffness in the contact phase for an elastic reaction of the structure with respect to the environment.

3 The PARAPLACER

One of the main drawbacks of parallel robots with closed kinematic chains is their poor ratio of workspace to installation space. The PARAPLACER (Fig. 2) is the first attempt to implement a parallel robot with the advantages of closed-loop structures such as high stiffness and high dynamics combined with a large optimized workspace.

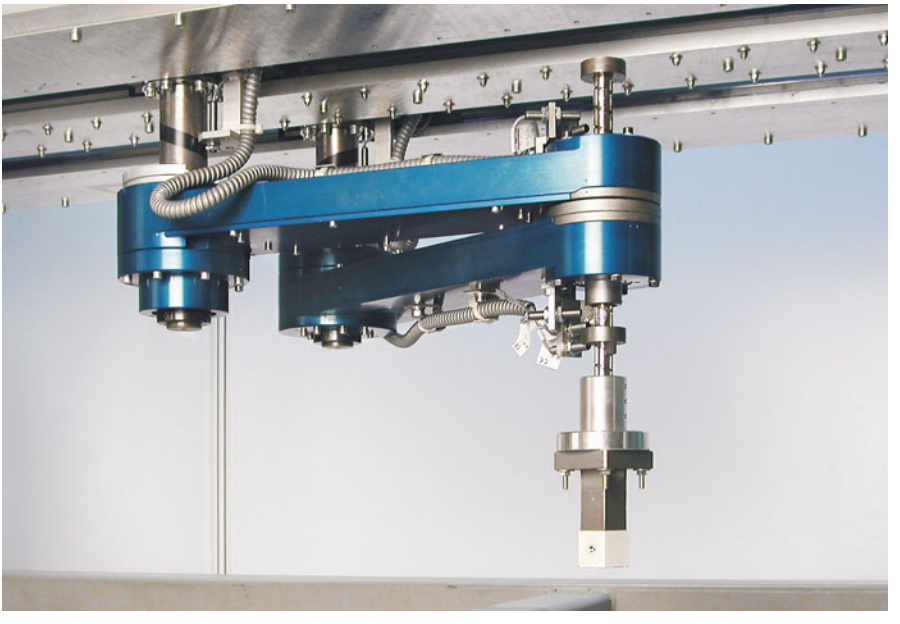


Fig. 2. Prototype of the PARAPLACER.

3.1 Design Features

The PARAPLACER is characterized by a PRRRP kinematic structure for x-y-movement and two additional serial components for φ and z motions. The PRRRP-structure is achieved by a planar closed kinematic chain consisting of one prismatic joint (P) followed by three revolute joints (R) and is closed by a second prismatic joint. The revolute joints are passive whereas the prismatic joints are active, actuated by two belt drives. This planar parallel part provides 2 DOF. The serial part of the hybrid manipulator adds two more DOF and is realized using a ball screw, which is attached in the center of the second revolute joint at point C (Fig. 3). The ball screw is actuated by synchronous belts and rotational drives (Fig. 4).

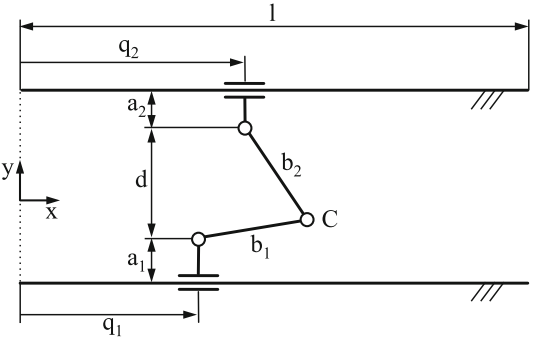


Fig. 3. Geometric parameters of the PARAPLACER.

As the robot provides four DOF, it is well suited for handling and assembly operations. A new approach has been developed that uses both direct kinematic problem (DKP) solutions to enlarge the workspace of the PARAPLACER. This means, the structure must pass through singularities of the second type. Therefore the kinematic parameters (Tab. 3) must be selected in a way that driving through singularities within the robot workspace is possible. Furthermore, the links which are connected at point *C* must lie in different planes such that no collision can occur.

Table 3. Geometric parameters of the PARAPLACER-structure.

Parameter description	**Symbol**	**Value**
Dist. between both revolute joints at linear drives	d	200 mm
Dist. between linear guides and revolute joint in x-y-plane	a_i	0 mm
Bar length	b_i	420 mm
Length of the linear guides	l	1400 mm
Possible linear actuator positions	$[q_{i,min}, q_{i,max}]$	[0, 1400] mm

3.2 Performance Characteristics

In Table 4 the performance data of the PARAPLACER are presented. It indicates that this parallel robot features a comparative ratio of workspace to installation space.

3.3 Fields of Research

In addition to the two configurations of the DKP, there are four other configurations of the inverse kinematic problem (IKP) of the PARAPLACER-structure. Figure 5 shows all 6 real configurations with their workspaces.

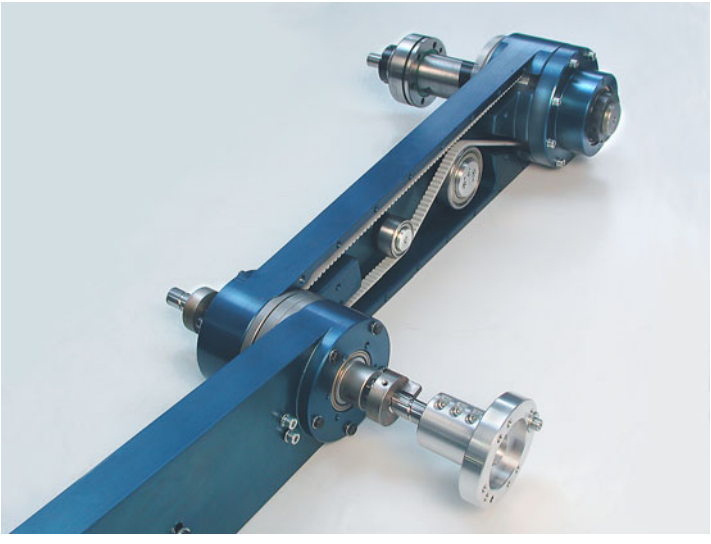


Fig. 4. Synchronous belts of the PARAPLACER ball screw.

Table 4. Performance data of the PARAPLACER.

Description	Data
Degree of freedom	2+2
Workspace x,y,z	1000 x 600 x 25 mm^3
Orientation z	180 °
Max. payload	3 kg
Max. velocity	3 m/s
Max. acceleration	2 g

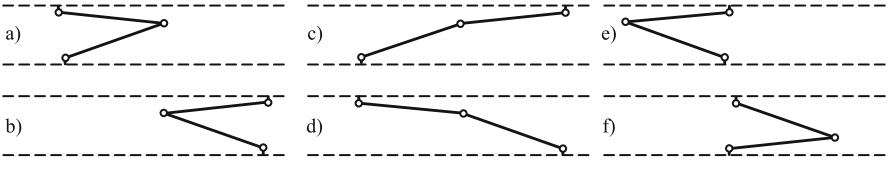


Fig. 5. Solutions of the IKP (a-d) and of the DKP (e-f).

Since these workspaces can be combined to one total workspace as shown in Fig. 6 the entire workspace can be substantially increased. To drive through singularities of the second type, a special algorithm has been implemented within the

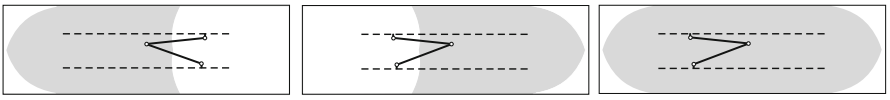


Fig. 6. Subworkspaces and total workspace of the PARAPLACER.

robot control. It is necessary to pass this point with specially secured motion since the structure gains one extra DOF in a singular pose. This can be realized by switching off one drive while the structure approaches a singularity. The inertia of the system forces the end-effector through the singularity into the other workspace configuration. Thereafter the inactive drive is activated again and the other part of the workspace can be used [9, 10].

4 The HEXAII-Robot

The Hexa-structure was originally presented by Pierrot [11]. At the SFB 562 it is used as a test bench for the development of new methods as well as components to demonstrate the dynamic capabilities of parallel structures.

4.1 Design Features

A prototype of the HEXAII-structure shown in Fig. 7 has been constructed by the SFB 562. The HEXAII is a six DOF manipulator composed of six equally designed kinematic chains, which connect a base platform with an end-effector platform. Each chain can be described as a serial RUS chain, where R stands for revolute, U for universal and S for spherical joint, respectively. The revolute joints are the active joints; the six actuators are rotational drives mounted under the base platform while all other joints remain passive.

Attributes of parallel structures are highly dependent on the dimensions and the arrangement of the kinematic chains comprising the structure. For instance, arranging the drives of the HEXAII-structure's six kinematic chains in three pairs of two drives with collinear rotational axes, allows singular positions on the z-axis of the base coordinate system **O** to be avoided (Fig. 8).

In these singular positions the structure would gain extra DOF which are not controllable by the drives. To determine the structure's dimension the fundamental kinematic parameters were systematically varied using a simulation tool and have been subsequently optimized with respect to the size of the workspace [12]. The parameters chosen for the HEXAII are given in Tab. 5.

The masses of the moved parts were minimized throughout the design process in order to tap the full potential of the HEXAII-structure's base mounted drives. Consequently, a total moved mass of 15.6 kg could be realized. In particular this aim applies to the integrated joints, which were designed at the Institute for Engineering Design of the Technische Universität Braunschweig. These joints are

Fig. 7. HEXAII of the SFB 562.

Table 5. Geometric parameters of the HEXAII.

Parameter description	**Symbol**	**Value**
z-coordinate of base coordinate system	h	0 mm
Base radius	r_{B_i}	360 mm
End-effector radius	r_{P_i}	51.96 mm
Distance between two parallel drives	$2b$	103.2 mm
Crank length	r_i	240 mm
Rod length	d_i	564 mm
Tool length x-component	$x_{\mathbf{T}}$	0 mm
Tool length y-component	$y_{\mathbf{T}}$	0 mm
Tool length z-component	$z_{\mathbf{T}}$	-120 mm

designed according to a design methodology which is specifically oriented towards the development of parallel structures [8, 13].

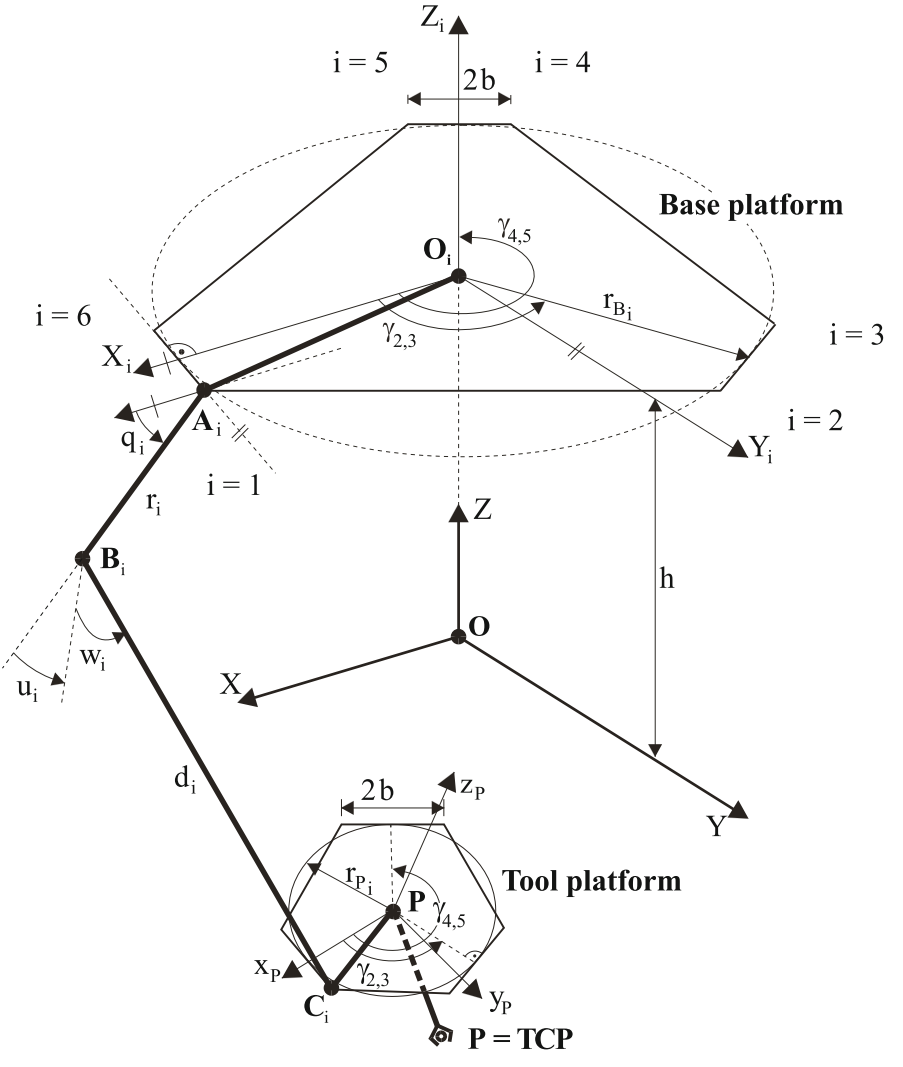


Fig. 8. Geometric parameters of the HEXAII.

In order to dimension the drives, the HEXAII-structure has been modeled using the multi-body simulation tool SIMPACK. Subsequently robot movements along predefined trajectories have been simulated to determine the necessary torques. Based on the results of these simulations, rotational direct drives with a continuous torque of 50 Nm and a peak torque of 177 Nm have been chosen. Due to the direct drive concept no gears are necessary. Thus the gears' inertia and clearance have been eliminated compared to a predecessor HEXAI-robot driven by geared motors. Furthermore, the direct drives feature high stiffness due to the direct coupling of the load allowing for high gain factors in the robot control.

4.2 Performance Characteristics

In Table 6 the reachable dynamics, the workspace dimensions, and the usable payload are listed.

Table 6. Performance data of the HEXAII.

Description	Data
Degree of freedom	6
Workspace x,y,z	⊘ 500 x 40 mm^3
Orientation z,y,x	20, 15, 15 °
Max. payload	3.5 kg
Max. velocity	5 m/s
Max. acceleration	6 g (10 g for a payload of 1 kg)

4.3 Fields of Research

As already noted the prototype of the HEXAII is used as a test bench for several interdisciplinary research topics. In addition to the structural characteristics the control system is of high relevance with respect to the performance of the HEXAII. Compared to serial robots, parallel structure control requires some additional functionalities. Therefore, a new control architecture was designed with a new middleware which also takes the real-time requirements into account [14, 15, 16, 17, 18].

The HEXAII-structure features singularities inside the workspace. Thus, for safety operation it is necessary that these areas can be detected before the HEXAII-structure gains extra DOF. For this purpose an index is elaborated and implemented in the control system indicating the closeness to the nearest singularities [19, 20, 21]. This index is further used for a special integrated path planning algorithm, which is inspired by the ant colony behavior [22].

In most cases, industrial robots are programmed by teach-in, which is a time-consuming process. A new approach uses skill primitives which are sensor-based, task-oriented robot commands for automated robot programming [23, 24, 25].

The HEXAII is calibrated to increase its absolute accuracy with different approaches [26, 27]. Therefore it is equipped with a new integrated joint sensor system which allows a calibration without external measurement systems [28, 29].

5 The TRIGLIDE-Robot

The prototype of the TRIGLIDE shown in Fig. 9 is similar to the linear Delta-robot of Clavel [30]. It is not only the most powerful robot, but has also the largest workspace of all developed SFB 562 structures.

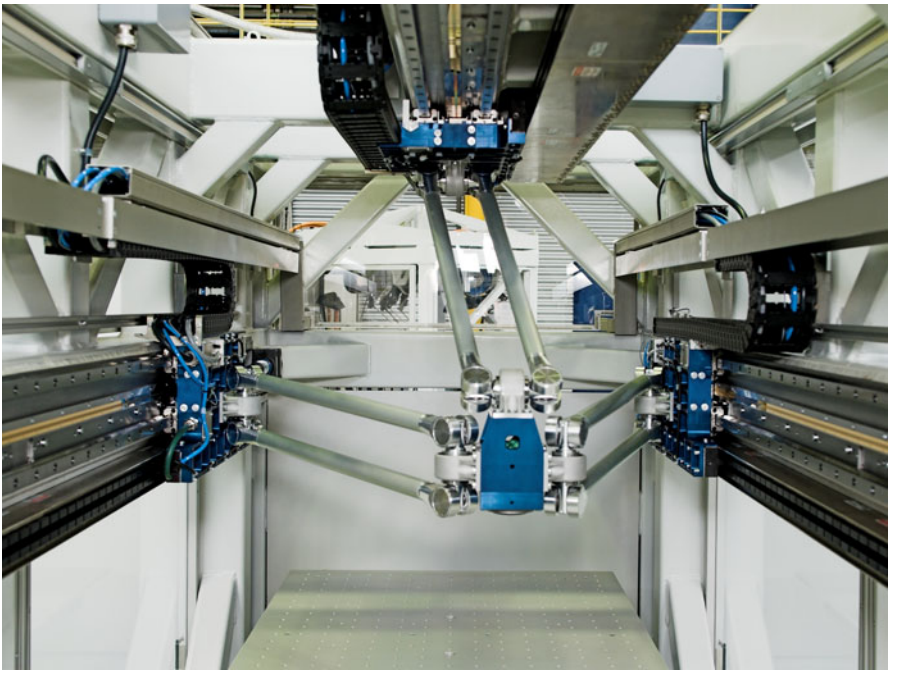


Fig. 9. TRIGLIDE of the SFB 562.

5.1 Design Features

The structure of the TRIGLIDE consists of three kinematic chains and is driven by linear direct actuators. The design with its parallel arrangement of the motors is similar to the machine tool Quickstep [31].

The kinematic chains are implemented as parallelograms which connect the motors with the end-effector platform. Thus, the orientation of the end-effector platform is constant. The parallel structure therefore has only translational DOF. For an additional rotation around the z-axis a serial axis is added to the end-effector platform, which makes the whole structure a hybrid system consisting of a parallel part and a serial part. The four DOF make classical Schönflies motions possible, which are necessary for efficient handling and assembly applications. Like the PARAPLACER, the TRIGLIDE can be used in two different workspace configurations to increase the ratio of workspace to installation space.

The specific optimized structure also allows to pass through singularities. Figure 10 shows the kinematic parameters of the structure. The values of the kinematic parameters, listed in Tab. 7, are chosen in a way to avoid collisions between the moving structure and the base, especially when passing through singular positions. This structure is also designed such that the kinematic chains allow the passing of all singularities of the IKP without encountering any collision between parts of the moving structure and without the occurrence of joint singularities in the chains.

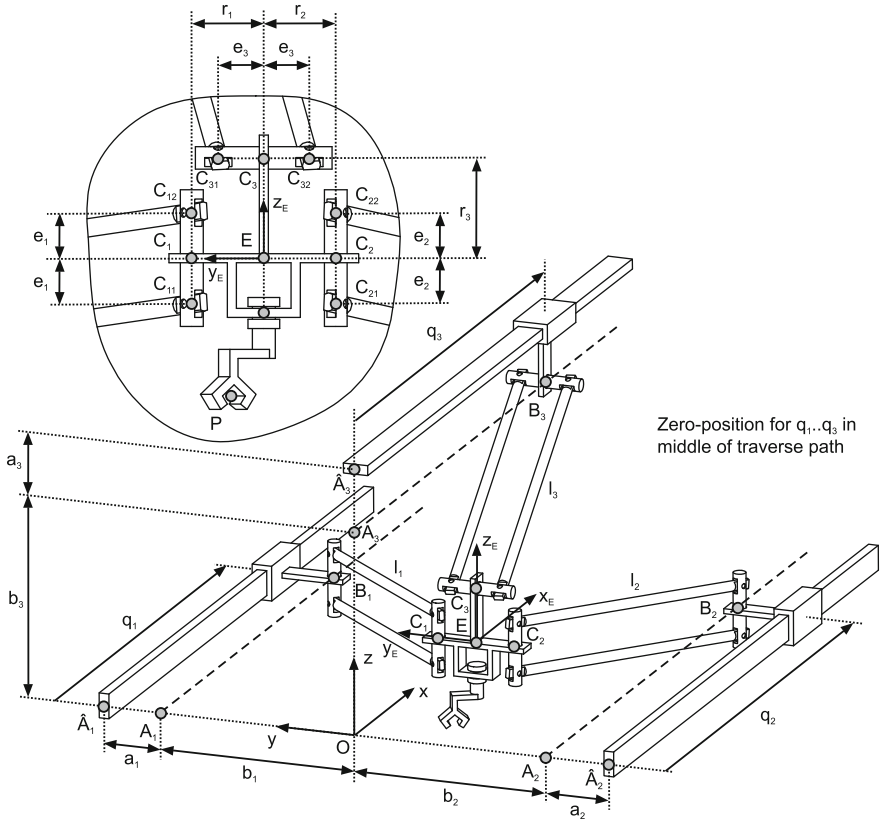


Fig. 10. Geometric Parameters of the TRIGLIDE-robot.

Table 7. Geometric parameters of the TRIGLIDE-structure.

Parameter description	**Symbol**	**Value**
Parallelogram length	l_i	600 mm
Dist. between parallelogram's cross bar to linear guides	a_i	105 mm
Dist. between base coord. system to lower linear guides	$b_1 = b_2$	379 mm
Dist. between base coord. system to upper linear guiding	b_3	415 mm
Dist. between platform coord. system to lower cross bars	$r_1 = r_2$	79 mm
Dist. between platform coord. system to upper cross bar	r_3	110 mm
Dist. between two joints of parallelogram's cross bars	$2e_i$	2x50 mm
Possible linear actuator positions	$[q_{i,min}, q_{i,max}]$	[-640, 640] mm

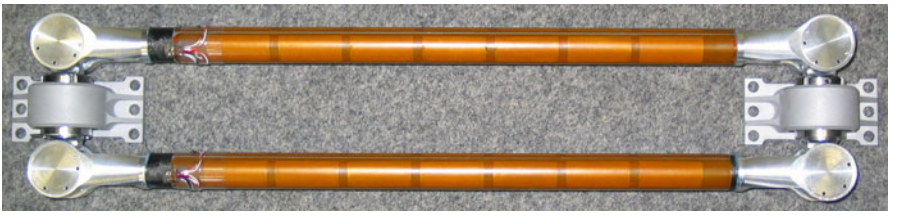


Fig. 11. Kinematic leg of the TRIGLIDE containing self-developed joints and adaptive links.

5.2 Performance Characteristics

Table 8 displays the attainable dynamics, the workspace dimensions and the usable payload of the prototype of the TRIGLIDE.

Table 8. Performance data of the TRIGLIDE.

Description	Data
Degree of freedom	3+1
Workspace x,y,z	1360 x 500 x 470 mm^3
Orientation z	180 °
Max. payload	3 kg
Max. velocity	5 m/s
Max. acceleration	10 g

5.3 Fields of Research

In order to enlarge the workspace, an approach is developed which enables the switch between working and assembly mode [10, 32]. A method has been implemented so that a statement about the current configuration of the TRIGLIDE-structure can be made [33]. These control functionalities are implemented, similar to the control based upon the system used for the HEXAII [14, 15, 16, 17, 18]. Another research area concerning the TRIGLIDE is to actively reduce structure vibrations resulting from very high dynamics. To this end an adaptronic system has been integrated into the structure and the control system. For active damping, without increasing the accelerated masses, the links of the structure are replaced by links with integrated piezoceramic foils (Fig. 11) [7, 34]. The conformity with a model-based control guarantees that the structure shows autonomic adaptive characteristics and can adjust itself to different conditions [4, 35].

6 Conclusions

In this contribution the four main parallel robot prototypes developed during ten years of research of the SFB 562 have been shown. In addition the key design features, the performance characteristics and the background for building these structures have been presented. This article gives a survey of the research activities which have been carried out by the members of the participating institutes.

Further informations and publications about these parallel robots and other fields of research of the SFB 562 can be found elsewhere in this book and on the corresponding website www.tu-braunschweig.de/sfb562.

Acknowledgements. Research and implementation of the prototypes of the parallel robots reported in this contribution were supported by the German Research Foundation (DFG) within the scope of the Collaborative Research Center SFB 562.

References

1. Algermissen, S., Rose, M., Sinapius, M., Stachera, K.: Robust gain-scheduling control for parallel robots with smart-structure components. In: Schütz, D., Raatz, A., Wahl, F.M. (eds.) Proceedings of the 3rd International Colloquium of the SFB 562: Robotic Systems for Handling and Assembly. Fortschritte in der Robotik, vol. 14, pp. 191–205. Shaker Verlag, Braunschweig (2008)
2. Kolbus, M., Wobbe, F., Algermissen, S., Stachera, K., Sinapius, M.: Sliding mode control of a parallel robot with robust vibration control. In: The 9th International Conference on Motion and Vibration Control (MoViC 2008), Munich, Germany (2008)
3. Stachera, K., Schreiber, F., Schumacher, W.: Modeling, Control, and Evaluation of an Experimental Adaptronic Five-Bar Robot. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 125–142. Springer, Heidelberg (2010)
4. Algermissen, S., Sinapius, M.: Robust Gain Scheduling for Smart-Structures in Parallel Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 159–174. Springer, Heidelberg (2010)
5. Pavlovic, N., Keimer, R., Franke, H.J.: Adaptronic revolute joints for parallel robots based on simultaneous quasi-statical axial and radial clearance adjustment. In: Proceedings of the ASME International Design Engineering Technical Conference & Computers and Information in Engineerung Conference (IDETC/CIE), San Diego, USA (2008)
6. Keimer, R., Sinapius, M.: Adaptive components for parallel robots. In: Schütz, D., Raatz, A., Wahl, F.M. (eds.) Proceedings of the 3rd International Colloquium of the SFB 562: Robotic Systems for Handling and Assembly. Fortschritte in der Robotik, vol. 14, pp. 181–190. Shaker Verlag, Braunschweig (2008)
7. Keimer, R., Sinapius, M.: Design and Implementation of Adaptronic Robot Components. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 413–427. Springer, Heidelberg (2010)
8. Pavlović, N., Otremba, R., Inkermann, D., Franke, H.J., Vietor, T.: Passive and Adaptive Joints for Parallel Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 429–444. Springer, Heidelberg (2010)

9. Helm, M.B.: Durchschlagende Mechanismen für Parallelroboter. In: Schriftenreihe des Instituts für Werkzeugmaschinen und Fertigungstechnik der TU Braunschweig. Shaker Verlag, Essen (2003) ISBN 3-8027-8671-8
10. Budde, C., Helm, M., Last, P., Raatz, A., Hesselbach, J.: Configuration Switching for Workspace Enlargement. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 175–189. Springer, Heidelberg (2010)
11. Pierrot, F., Uchiyama, M., Dauchez, P., Fournier, A.: A new design of a 6-DOF parallel robot. Journal of Robotics and Mechatronics 2(4), 92–99 (1990)
12. Frindt, M., Krefft, M., Hesselbach, J.: Structure and Type Synthesis of Parallel Manipulators. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 17–37. Springer, Heidelberg (2010)
13. Otremba, R., Jänicke, T., Franke, H.-J.: Systematische Entwicklung passiver Gelenke für Parallelstrukturen. VDI-Berichte Nr. 1679, 545–550 (2002)
14. Maaß, J., Kolbus, M., Bier, C., Wobbe, F., Schumacher, W., Raatz, A., Hesselbach, J.: Advances in motion control for high-performance parallel robots. In: Schütz, D., Raatz, A., Wahl, F.M. (eds.) Proceedings of the 3rd International Colloquium of the SFB 562: Robotic Systems for Handling and Assembly. Fortschritte in der Robotik, vol. 14, pp. 101–112. Shaker Verlag, Braunschweig (2008) ISBN 978-3832271299
15. Dietrich, F., Maaß, J., Raatz, A., Hesselbach, J.: RCA562: Control Architecture for Parallel Kinematic Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 315–331. Springer, Heidelberg (2010)
16. Michalik, H., Dadji, Y., Maaß, J.: A Communication Architecture for Distributed Real-Time Robot Control. In: Schütz, D., Raatz, A., Wahl, F.M. (eds.) Proceedings of the 3rd International Colloquium of the SFB 562: Robotic Systems for Handling and Assembly. Fortschritte in der Robotik, vol. 14, pp. 127–140. Shaker Verlag, Braunschweig (2008)
17. Dadji, Y., Michalik, H., Kohn, N., Steiner, J., Beckmann, G., Möglich, T., Varchmin, J.U.: A Communication Architecture for Distributed Real-Time Robot Control. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 213–231. Springer, Heidelberg (2010)
18. Finkemeyer, B., Kröger, T., Wahl, F.M.: A Middleware for High-Speed Distributed Real-Time Robotic Applications. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 193–212. Springer, Heidelberg (2010)
19. Bier, C.: Geometrische und physikalische Analyse von Singularitäten bei Parallelstrukturen. In: Schriftenreihe des Instituts für Werkzeugmaschinen und Fertigungstechnik der TU Braunschweig. Vulkan Verlag, Essen (2006) ISBN 978-3-8027-8690-7
20. Bier, C., Campos, A., Hesselbach, J.: Direct singularity closeness indexes for the Hexa parallel robot. In: Lennarčič, J., Roth, B. (eds.) Advances in Robot Kinematics, pp. 239–246. Springer, Dordrecht (2006) ISBN 978-1-4020-4940-8
21. Dietrich, F., Maaß, J., Bier, C., Pietsch, I., Raatz, A., Hesselbach, J.: Detection and Avoidance of Singularities in Parallel Kinematic Machines. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 77–92. Springer, Heidelberg (2010)
22. Maaß, J., Raatz, A., Hesselbach, J., Künning, G.: Partikelgestützte Bahnplanung zur Singularitätsvermeidung für Parallelkinematiken. In: Robotik, VDI, Munich, Germany (2008)
23. Thomas, U., Maaß, J., Wahl, F.M., Hesselbach, J.: Towards a new concept of robot programming in high speed assembly applications. In: IEEE International Conference on Intelligent Robotic Systems (IROS), Edmonton, Canada, pp. 3932–3938 (2005)

24. Thomas, U., Wahl, F.M.: Assembly Planning and Task Planning: Two Prerequisites for Automated Robot Programming. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 333–354. Springer, Heidelberg (2010)
25. Kröger, T., Finkemeyer, B., Wahl, F.M.: Manipulation Primitives: A Universal Interface Between Sensor-Based Motion Control and Robot Programming. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 293–313. Springer, Heidelberg (2010)
26. Last, P., Budde, C., Bier, C., Hesselbach, J.: HEXA-parallel-structure calibration by means of angular passive joint sensors. In: Proceedings of the IEEE International Conference on Mechatronics and Automation, Niagara Falls, Canada, pp. 1300–1305 (2005)
27. Last, P., Raatz, A., Hesselbach, J.: Calibration of Parallel Kinematic Structures - Overview, Classification and Comparison. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 93–106. Springer, Heidelberg (2010)
28. Büttgenbach, S., Güttler, J., Last, P., Bier, C., Otremba, R.: Development of angular joint-sensors and application to parallel robots. In: Last, P., Budde, C., Wahl, F.M. (eds.) Proceedings of the 2nd International Colloquium of the SFB 562: Robotic systems for handling and assembly. Fortschritte in der Robotik, vol. 9, pp. 237–251. Shaker Verlag, Braunschweig (2005) ISBN 3-8322-3866-2
29. Kirchhoff, M.R., Güttler, J., Wogersien, A., Pavlović, N., Otremba, R., Franke, H.J., Büttgenbach, S.: Design and Implementation of New Sensors and their Integration in Joints. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 445–459. Springer, Heidelberg (2010)
30. Clavel, R.: Robots parallèles. Techniques de l'Ingénieur, traité Mesures et Contrôle 7(7710), 1–8 (1994)
31. Bleicher, F.: Optimizing a three-axes machine-tool with parallel kinematic structure. In: Proceedings of the 3rd Chemnitz Parallel Kinematics Seminar, Chemnitz, Germany, pp. 883–894 (2002)
32. Budde, C., Last, P., Hesselbach, J.: Development of a Triglide-robot with enlarged workspace. In: IEEE International Conference on Robotics and Automation, Roma, Italy, pp. 543–548 (2007)
33. Budde, C., Rose, M., Maaß, J., Raatz, A.: Automatic detection of assembly mode for a Triglide-robot. In: IEEE International Conference on Robotics and Automation, Pasadena, USA, pp. 1568–1575 (2008)
34. Keimer, R., Algermissen, S., Pavlovic, N., Budde, C.: Smart structures technologies for parallel kinematics in handling and assembly. In: Proceedings of SPIE Smart Structures and Materials/NDE, San Diego, California (2007)
35. Algermissen, S., Keimer, R., Rose, M., Breitbach, E.: Applied robust control for vibration suppression in parallel robots. In: Proceedings of the 22nd International Symposium on Automation and Robotics in Construction (ISARC), Ferrara, Italy (2005)

Modeling, Control, and Evaluation of an Experimental Adaptronic Five-Bar Robot

Krzysztof Stachera, Frank Schreiber, and Walter Schumacher

Abstract. Embedded actuators offer a promising approach to counteract unwanted vibrations arising in the structure of light-weight parallel manipulators. These actuators and the elasticity of the robot structure introduce additional degrees of freedom, increasing the mathematical and computational effort in the description of the robot kinematics and dynamics.

In this paper computationally efficient algorithms for the calculation of the Jacobian and the direct dynamics of parallel robots are presented which explicitly exploit specific properties of parallel structures. The new concepts have been applied to the FIVE-BAR planar parallel manipulator and evaluated by comparison with conventional techniques. Based upon the presented methods, a model-based robust control strategy has been developed and validated using the experimental results obtained from the implementation for the FIVE-BAR manipulator.

1 Introduction

Modern industrial processes, like handling and assembly tasks in the electronics industry, require high performance manipulators, especially regarding the obtainable speed and precision. Parallel manipulators have the potential to meet some of those goals. As opposed to their omnipresent serial counterparts, the drives of parallel mechanisms are mounted to the base or at least close to the base allowing for a significant reduction of the moved structure masses and at the same time increasing the obtainable precision of the parallel manipulators. Since the heavy drives are no longer moved, the links and joints can also be designed lighter accordingly. The resulting light-weight manipulators are able to realize higher accelerations and velocities, but tend to suffer from a reduced damping which may lead to unwanted vibrations during and after the execution of motions [1]. Since following steps of

Krzysztof Stachera · Frank Schreiber · Walter Schumacher
Technische Universität Braunschweig, Institute of Control Engineering,
Hans-Sommer-Straße 66, 38106 Braunschweig, Germany
e-mail: {stachera,schreiber,w.schumacher}@ifr.ing.tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 125–142.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

a handling and assembly task require the decay of the end-effector oscillations below a certain level, they increase the time of one working cycle. To allow for the settling of the end-effector some of the time gained by performing faster motions is unavoidably lost. In order to fully utilize the potential of the elastic light-weight parallel manipulators for high speed applications, methods for vibration suppression were developed, which will be presented in this contribution.

2 Related Works

Elasticities of the robot structure induce deformations of the structure and vibrations of the end-effector during and after movements. The singular perturbation method allows to separate the motion control of the manipulator from the vibration suppression control for the structure since both take place at different time scales [2, 3]. Using the robot drives for the purposes of vibration control can be complicated for parallel elastic manipulators [4, 5, 6]. As opposed to their serial counterparts, the drives of parallel manipulators are not always connected directly to the elastic elements of the structure, thus their influence upon the elastic modes is often reduced.

These problems can be overcome by using so-called "smart materials" which, in combination with appropriate control algorithms, form the field of Adaptronics. Among those "smart materials" are piezo actuators which can be manufactured in diverse shapes [7] and thus be easily integrated into the robot structure. Furthermore, the placement can be optimized in order to achieve a maximum authority over the elastic modes [8, 9]. The separate control of the elastic structure using piezo patches and of the robot motion using conventional drives has been studied for several serial kinematics [2]. It was shown that the control algorithms for the robot motion and for the structure's vibrations can be designed independently [10, 11, 12, 13]. However, the control concepts for elastic parallel manipulators have not yet been studied as thoroughly as in the field of serial robots. This provided the motivation for the SFB 562 to conduct research in this area. Within this project two different control strategies have been developed. One approach employs a gain-scheduling algorithm to generate the controller output from a set of robust controllers, each designed for an identified locally valid linear model [14]. The second approach, presented hereafter, was to develop a model of the nonlinear manipulator dynamics, which is valid over the manipulator's workspace and allows the computation under real-time conditions. Consequently, a model-based controller was designed in order to ensure a constant controller performance in every manipulator position.

3 Dynamics of Elastic Parallel Manipulators

Several techniques from the field of mechanics can be employed for deriving the typically nonlinear models of the dynamics of parallel manipulators. One elegant and efficient method, which is widely used in the area of parallel manipulators, is the *Lagrangian method* [15, 16]. For parallel manipulators, additional

equations describing the closed kinematic loop constraints must be provided. In the Lagrange-D'Alembert (L-D'A) formulation the Jacobians of the kinematic constraints parameterized by the non-redundant coordinates are used for this purpose [17, 18]. The approach using the Jacobians allows to divide the parallel manipulator into several serial kinematic chains in the modeling process. Consequently, well known methods and techniques, which were already applied to serial elastic robots, can thus be used for the description of the manipulator's chain dynamics [19, 20].

The following derivation of the equations of motion of parallel manipulators in Sect. 3.2 is based upon the L-D'A formulation. It provides compact equations of the manipulator's dynamics which are advantageous for system analysis and the following control design [21, 22]. In addition to this method, a new method for the derivation of the Jacobians of parallel manipulators is introduced [23] in Sect. 3.3. In Sects. 3.4 and 3.5 a new computationally efficient method for the simultaneous/-distributed calculation of its direct dynamics (SCDD) will be presented and verified.

3.1 Description of Elasticities in the Manipulator's Structure

The links of a parallel manipulator can be regarded as the main source of vibrations in the robot structure since they are mainly built as long and slender elements [24]. In cases in which forces are exclusively applied to the end points of simple continuous link elements their deformations can be described by partial differential equations or using the Ritz method. For more complex continua and load cases the finite element method (FEM) allows an approximative description, providing a system of ordinary differential equations in the form

$$\boldsymbol{M_i}\ddot{\boldsymbol{q}}_{\boldsymbol{ie}}(t) + \boldsymbol{D_i}\dot{\boldsymbol{q}}_{\boldsymbol{ie}}(t) + \boldsymbol{K_i}\boldsymbol{q}_{\boldsymbol{ie}}(t) = \boldsymbol{0} \tag{1}$$

with $\boldsymbol{M_i} \in \mathbb{R}^{n\times n}$ representing a symmetrical positive definite mass or inertia matrix, $\boldsymbol{D_i}$ an attenuation matrix, $\boldsymbol{K_i} \in \mathbb{R}^{n\times n}$ a symmetrical positive definite or positive semidefinite stiffness matrix, and $\boldsymbol{q_{ie}}(t) \in \mathbb{R}^n$ the vector of the coordinates of the elastic deformations in the nodes [25, 26].

One drawback of the FEM, however, is the high order of the obtained matrices which might prohibit a real-time calculation of the equations of motion. In the field of robotics the method of concentrated parameters is therefore applied to derive reduced order models of the form (1). Although a reduction of the obtainable precision has to be taken into account the resulting model with discrete elastic degrees of freedom is applicable for most robot architectures [19, 27, 28].

3.2 Derivation of the Equations of Motion

The concentrated parameter model of an elastic parallel manipulator forms a system in which the coordinates of the active, passive and elastic degrees of freedom are interdependent due to constraints. For deriving the dynamic equations this

system is transformed into a representation using independent generalized coordinates defined in such a way that the motion constraints are implicitly fulfilled [21, 22]. The algorithm to derive the equations of motion in these coordinates is detailed below.

3.2.1 Inverse Dynamics

The procedures for the derivation of the inverse dynamics of parallel manipulators correspond to the methods which are known for serial manipulators and consist of three steps [21]:

1. *Transformation of the System*: Each closed kinematic loop of the parallel manipulator is separated at a passive joint, end-effector or link. This results in a reduced system with a tree structure. Consequently, only serial kinematic chains can be found in this system. Furthermore it is assumed that all remaining passive joints possess virtual actuators.
2. *Computation of the Torques*: The torques and forces of the real and virtual actuators are computed for each kinematic chain. These torques and forces cause a movement in every chain, and these movements correspond to the movement of the original closed-link structure.
3. *Transformation of the Torques*: The torques and forces of the original parallel manipulator's actuators are calculated from the forces and torques of the tree structure by considering the additional closed kinematic loop constraints.

It is assumed that the manipulator consists of l closed kinematic loops and possesses a total of $e + a + p = n$ one-DOF joints, with e being the number of discrete elastic DOF, a the number of active and p the number of passive joints. (See Fig. 4b for a FIVE-BAR example.) The coordinates of the active joints $\boldsymbol{q_a}$ and of the discrete elastic joints $\boldsymbol{q_e}$ form a set of non-redundant coordinates. The controllability of the manipulator in absence of elasticities is assumed. According to the first step this system is divided into a tree structure. The dimensions of $\boldsymbol{q_a}$ and $\boldsymbol{q_e}$ remain identical to those of the original structure ($n_a = (n - p - e)$ and $n_e = e$ respectively). The number of passive joint coordinates $\boldsymbol{q_p}$ amounts to $n_p = (p - l)$. Thus the coordinates of the tree structure are:

$$\boldsymbol{q_t} = \boldsymbol{q_t}(\boldsymbol{q_a}, \boldsymbol{q_p}, \boldsymbol{q_e}), \tag{2}$$

where $\boldsymbol{q_a} \in \mathbb{R}^{n_a}$, $\boldsymbol{q_p} \in \mathbb{R}^{n_p}$, $\boldsymbol{q_e} \in \mathbb{R}^{n_e}$ and $\boldsymbol{q_t} \in \mathbb{R}^{n_t}$, with $n_t = n_a + n_p + n_e$. The redundant coordinates of the passive joints $\boldsymbol{q_p}$ are determined by the coordinates of the active joints $\boldsymbol{q_a}$ and the elastic DOF $\boldsymbol{q_e}$

$$\boldsymbol{q_p} = \boldsymbol{q_p}(\boldsymbol{q_{ae}}), \tag{3}$$

where $\boldsymbol{q_{ae}} \in \mathbb{R}^{n_{ae}}$ and $n_{ae} = n_a + n_e$. Generally, the relation represented in (3) may not always exist analytically, but a solution to determine the relationship between the velocities and accelerations of the active and passive joints can always

be found [29, 30]. For this purpose the closed kinematic loop constraints $\boldsymbol{h}$ of the parallel manipulator are introduced:

$$\boldsymbol{h}(\boldsymbol{q_t}) = \boldsymbol{h}(\boldsymbol{q_{ae}}, \boldsymbol{q_p}) = 0. \tag{4}$$

After the differentiation of (4) a Pfaffian form of the constraints is obtained:

$$\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}_{ae}^T}\dot{\boldsymbol{q}}_{ae} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}_{p}^T}\dot{\boldsymbol{q}}_{p} = 0. \tag{5}$$

In order to parameterize the configuration space of the manipulator, a Jacobian of the kinematic constraints parameterized by the non-redundant coordinates has to be derived. For this purpose it will be assumed that the robot is normally actuated and away from actuator singularity. Therefore, matrix $\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}_p^T}$ is square and invertible. The configuration space of the manipulator can be smoothly parameterized by the coordinates of the active joints and the elastic DOF $\boldsymbol{q_{ae}}$:

$$\dot{\boldsymbol{q}}_p = -\left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}_p^T}\right)^{-1}\left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}_{ae}^T}\right)\dot{\boldsymbol{q}}_{ae} = \left(\frac{\partial \boldsymbol{q}_p}{\partial \boldsymbol{q}_{ae}^T}\right)\dot{\boldsymbol{q}}_{ae}. \tag{6}$$

Based on (6) the Jacobian of the parameterization matrix can be written:

$$\boldsymbol{W} = \frac{\partial \boldsymbol{q_t}}{\partial \boldsymbol{q}_{ae}^T}. \tag{7}$$

In accordance with the second step, the torques/forces of the tree-structure have to be calculated. They can be derived from the equations of motion in matrix form:

$$\boldsymbol{M_t}(\boldsymbol{q_t})\ddot{\boldsymbol{q}}_t + \boldsymbol{C_t}(\dot{\boldsymbol{q}}_t, \boldsymbol{q_t})\dot{\boldsymbol{q}}_t + \boldsymbol{\eta_t}(\boldsymbol{q_t}) + \boldsymbol{K_t}\boldsymbol{q_t} + \boldsymbol{D_t}\dot{\boldsymbol{q}}_t = \boldsymbol{\tau_t}, \tag{8}$$

with $\boldsymbol{\tau_t} \in \mathbb{R}^{n_t}$ representing all torques/forces of the real and virtual drives of the reduced system. $\boldsymbol{M_t}(q_t), \boldsymbol{C_t}(\dot{q}_t, q_t) \in \mathbb{R}^{n_t \times n_t}$ are the inertia matrix and the Coriolis matrix of the tree structure, respectively. $\boldsymbol{\eta_t}(\boldsymbol{q_t}) \in \mathbb{R}^{n_t}$ is the vector of the gravity force projected into joint space. $\boldsymbol{K_t} \in \mathbb{R}^{n_t \times n_t}$ and $\boldsymbol{D_t} \in \mathbb{R}^{n_t \times n_t}$ represent the diagonal matrices of the discrete elasticities and discrete damping elements in joint space. By using the matrix $\boldsymbol{W}$ from (7) the equations of the dynamics of the tree structure (8) can be transformed into the equations of the closed-link mechanism. Then, they are expressed only in dependence on the coordinates of the active joints $\boldsymbol{q_a}$ and the elastic DOF $\boldsymbol{q_e}$:

$$\boldsymbol{M_c}(\boldsymbol{q_t})\ddot{\boldsymbol{q}}_{ae} + \boldsymbol{C_c}(\dot{\boldsymbol{q}}_t, \boldsymbol{q_t})\dot{\boldsymbol{q}}_{ae} + \boldsymbol{\eta_c}(\boldsymbol{q_t}) + \boldsymbol{K_c}\boldsymbol{q_{ae}} + \boldsymbol{D_c}\dot{\boldsymbol{q}}_{ae} = \boldsymbol{\tau_c}, \tag{9}$$

with:

$$\boldsymbol{M_c} = \boldsymbol{W^T M_t W} \in \mathbb{R}^{n_{ae} \times n_{ae}}, \tag{10}$$

$$\boldsymbol{C_c} = \boldsymbol{W^T M_t \dot{W}} + \boldsymbol{W^T C_t W} \in \mathbb{R}^{n_{ae} \times n_{ae}}, \tag{11}$$

$$\boldsymbol{\eta_c} = \boldsymbol{W^T \eta_t} \in \mathbb{R}^{n_{ae}}, \tag{12}$$

$$\boldsymbol{K_c} = \boldsymbol{W^T K_t W} \in \mathbb{R}^{n_{ae} \times n_{ae}}, \tag{13}$$

$$\boldsymbol{D_c} = \boldsymbol{W^T D_t W} \in \mathbb{R}^{n_{ae} \times n_{ae}}, \tag{14}$$

$$\boldsymbol{\tau_c} = \boldsymbol{W^T \tau_t} \in \mathbb{R}^{n_{ae}}. \tag{15}$$

Proofs of these transformations and their derivations have been published in [21, 24]. The redundant coordinates of the passive joints, which are necessary for the computation of the compact matrices, result from the closed kinematic loop constraints of the parallel manipulator (3) as well as their first (6) and second derivatives.

Using this method, the equations of the dynamics of the tree structure (8) are transformed into the compact equations of the closed-link mechanism (9) and parameterized by the non-redundant coordinates $\boldsymbol{q_{ae}}$ (10)–(15). Consequently the drive torques can be calculated.

3.3 Derivation of the Jacobian of Parallel Manipulators

In conventional methods the Jacobian of a parallel manipulator is derived using the velocity vector-loop method [16] or by analyzing the parallel manipulator's statics [17, 29]. The L-D'A formulation allows to systematically convert between the single models of serial kinematic chains and the model of the compact parallel manipulator [22]. However, it has not yet been demonstrated how the Jacobians of the serial kinematic chains of the tree structure $\boldsymbol{J_i}$ can be transformed into the Jacobian of the parallel manipulator $\boldsymbol{G}$. An algorithm has been developed to perform this transformation. For this purpose, the $\boldsymbol{W}$ matrix representing the parameterization of the configuration space from (7) and the static matrix $\boldsymbol{S}$ of the parallel manipulator are used. The static matrix $\boldsymbol{S}$ describes the relationship between the forces in the rods of the parallel manipulator $\boldsymbol{f_b} \in \mathbb{R}^{n_a}$ and the force $\boldsymbol{f_x}$ on its end-effector e, shown in Fig. 1.

The forces in the branches $\boldsymbol{f_b}$ can be calculated using the following relationship:

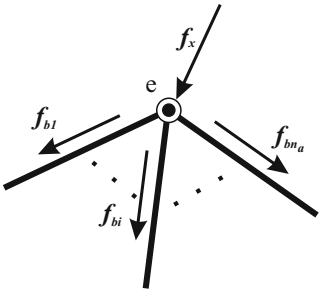


Fig. 1. Distribution of the Cartesian force on the end-effector $\boldsymbol{f_x}$ into the branch forces $\boldsymbol{f_{bi}}$ of the parallel manipulator's arms

$$\boldsymbol{f_b} = \left[\boldsymbol{s_1} \ \ldots \ \boldsymbol{s_{n_a}} \right]^{-1} \boldsymbol{f_x} = \boldsymbol{S}^{-1} \boldsymbol{f_x}, \tag{16}$$

If the matrix $\boldsymbol{S}$ is not square then $\boldsymbol{S^{-1}}$ is the pseudo-inverse $\boldsymbol{S^{+}}$. The elements of the $\boldsymbol{S}$-Matrix $\boldsymbol{s_i}(q_{ai}, \boldsymbol{q_{pi}}, \boldsymbol{q_{ei}})$ comprise the vector that relates $\boldsymbol{f_{bi}}$, the force of the i^{th} chain, and $\boldsymbol{f_{bxi}}$, the Cartesian force resulting from $\boldsymbol{f_{bi}}$.

$$\boldsymbol{f_{bxi}} = \boldsymbol{s_i} f_{bi}. \tag{17}$$

In matrix form with matrix $\boldsymbol{U}$ this relation leads to:

$$\boldsymbol{f_{bx}} = \begin{bmatrix} \boldsymbol{s_1} & \ldots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \ldots & \boldsymbol{s_{n_a}} \end{bmatrix} \boldsymbol{f_b} = \boldsymbol{U f_b}. \tag{18}$$

Now, the Jacobian $\boldsymbol{J_t}$ of the tree structure is introduced:

$$\boldsymbol{J_t} = \begin{bmatrix} \boldsymbol{J_1} & \ldots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \ldots & \boldsymbol{J_{n_a}} \end{bmatrix}, \tag{19}$$

where $\boldsymbol{J_i}(q_{ai}, \boldsymbol{q_{pi}}, \boldsymbol{q_{ei}})$ are the n_a-Jacobians of the serial kinematic chains. In order to eliminate the dependencies of the passive joint coordinates $\boldsymbol{q_p}$ for the calculation of the Jacobian $\boldsymbol{G}$ of a parallel manipulator, the matrix $\boldsymbol{J_t}$ is parametrized with the matrix $\boldsymbol{W}$. After this parametrization the new matrix does not yet represent the mapping between the joint and Cartesian velocities nor that between the joint torques and the end-effector forces of the parallel manipulator. In order to obtain this mapping, the matrices $\boldsymbol{S}$ and $\boldsymbol{U}$ are be introduced. Applying the transformations (16) and (18), the Jacobian of the parallel manipulator can be derived from the following relation:

$$\boldsymbol{G^{+T}} = \boldsymbol{W^T J_t^T U S^{-1}}. \tag{20}$$

This pseudo-inverse Jacobian $\boldsymbol{G^{+T}}$ represents the sought mapping between the Cartesian force $\boldsymbol{f_x}$ on the end-effector of the parallel manipulator and the forces/-torques $\boldsymbol{\tau_{ae}} \in \mathbb{R}^{n_{ae}}$ in the manipulator's structure in the joint space:

$$\boldsymbol{\tau_{ae}} = \boldsymbol{G^{+T} f_x} \tag{21}$$

The presented method has the significant advantage that the derivation of the serial Jacobians is less complex compared to the direct derivation of the compact Jacobian using standard methods.

3.4 Direct Dynamics

The equations of the direct dynamics are obtained from the compact equations (9) of the manipulator's inverse dynamics:

$$\ddot{\boldsymbol{q}}_{ae} = \boldsymbol{M}_c(\boldsymbol{q}_t)^{-1}(\boldsymbol{\tau}_c - \boldsymbol{C}_c(\dot{\boldsymbol{q}}_t, \boldsymbol{q}_t)\dot{\boldsymbol{q}}_{ae} - \boldsymbol{\eta}_c(\boldsymbol{q}_t) - \boldsymbol{K}_c\boldsymbol{q}_{ae} - \boldsymbol{D}_c\dot{\boldsymbol{q}}_{ae}), \tag{22}$$

This method consequently produces the compact equations of the direct dynamics of the elastic parallel manipulator. The disadvantage is that the computations cannot be executed in parallel. If more parameters need to be considered for the modeling, e.g. Finite-Element-Method [31], the matrices of the compact system may no longer be feasibly calculated in real-time due to their size and complexity.

3.4.1 Simultaneous Calculation of the Direct Dynamics

The calculations of the direct dynamic equations in the compact form derived in Sect. 3.4 are computationally complex and may lead to complications on systems with short cycle times. In order to overcome this limitation, a new method for the simultaneous/distributed calculation of the direct dynamics (SCDD) of the elastic parallel manipulator is suggested [23, 32]. In this method, the parallel manipulator is handled as a reduced system similar to the approach used for the inverse dynamics of the L-D'A formulation. However, in this system the closed kinematic loop constraints of the elastic parallel structure are represented by forces and torques, just as in the case of the Lagrangian equations of the first type [33, 16].

The equations of the tree structure have the known form shown in (8). These equations will now be factored into the equations of motion of the individual serial kinematic chains:

$$\boldsymbol{M}_{ti}(\boldsymbol{q}_{ti})\ddot{\boldsymbol{q}}_{ti} + \boldsymbol{C}_{ti}(\dot{\boldsymbol{q}}_{ti}, \boldsymbol{q}_{ti})\dot{\boldsymbol{q}}_{ti} + \boldsymbol{\eta}_{ti}(\boldsymbol{q}_{ti}) + \boldsymbol{J}_i^T\boldsymbol{f}_{bxi} + \boldsymbol{K}_{ti}\boldsymbol{q}_{ti} + \boldsymbol{D}_{ti}\dot{\boldsymbol{q}}_{ti} = \boldsymbol{\tau}_{ti}, \tag{23}$$

for $i = 1 \ldots n_a$, where i denotes one kinematic chain with the associated variables $\boldsymbol{q}_{ti}^T = \left[q_{ai}\ \boldsymbol{q}_{pi}^T\ \boldsymbol{q}_{ei}^T\right]$ and torques $\boldsymbol{\tau}_{ti}^T = \left[\tau_{ai}\ \boldsymbol{\tau}_{pi}^T\ \boldsymbol{\tau}_{ei}^T\right]$ of the active τ_{ai} and passive $\boldsymbol{\tau}_{pi}$ joints and additional structure torques $\boldsymbol{\tau}_{ei}$. The force $\boldsymbol{f}_{bxi}$ represents an external force acting upon the ends of the i^{th}-branches of the tree structure. The equations of the direct dynamics of each of these chains can then be formulated:

$$\ddot{\boldsymbol{q}}_{ti} = \boldsymbol{M}_{ti}(\boldsymbol{q}_{ti})^{-1}\left(\boldsymbol{\tau}_{ti} - \boldsymbol{C}_{ti}(\dot{\boldsymbol{q}}_{ti}, \boldsymbol{q}_{ti})\dot{\boldsymbol{q}}_{ti} - \boldsymbol{\eta}_{ti}(\boldsymbol{q}_{ti}) - \boldsymbol{K}_{ti}\boldsymbol{q}_{ti} - \boldsymbol{D}_{ti}\dot{\boldsymbol{q}}_{ti} - \boldsymbol{J}_i^T\boldsymbol{f}_{bxi}\right), \tag{24}$$

for $i = 1 \ldots n_a$. Thus, the direct dynamics of each serial kinematic chain can be calculated. The inputs of each of these equations are the external forces acting upon the end of the particular serial kinematic chain $\boldsymbol{f}_{bxi}$ and the input torques τ_{ai} and $\boldsymbol{\tau}_{pi}$. The torques of the elastic DOF $\boldsymbol{\tau}_{ei}$ result from the material properties like stiffness and damping. The input of the tree-structure (8) and of the compact parallel manipulator (9) is the torque/force vector τ_c. According to the D'Alembert principle the performed virtual work for both systems, the reduced and the original one, has to be equal [21]. Moreover, the working conditions of both systems have to be equal [24]. This implies the following conditions:

$$\begin{aligned} \boldsymbol{\tau}_a &= \boldsymbol{\tau}_c, \\ \boldsymbol{\tau}_p &= \boldsymbol{0}. \end{aligned} \tag{25}$$

In order to fulfill these conditions, the new torques/forces of the closed-loop constraints acting upon the end of each i^{th}-branch must be calculated. Along with the drive torques they are applied to the partial equations of the direct dynamics (24).

The calculation of the torques/forces starts with the computation of the interdependent torques/forces of the tree-structure resulting from the input torque vector $\boldsymbol{\tau_c}$. The relation between them is established in (15). All the torques/forces of the passive joints of the tree-structure will be calculated. To perform this task those partial matrices and vectors of the equations of the reduced system (23) are taken, which are associated with the virtual torques of the passive joints:

$$\boldsymbol{\tau_{pi}} = \boldsymbol{M_{pij}}(\boldsymbol{q_{ti}})\ddot{\boldsymbol{q}}_{\boldsymbol{ti}} + \boldsymbol{C_{pij}}(\dot{\boldsymbol{q}}_{\boldsymbol{ti}}, \boldsymbol{q_{ti}})\dot{\boldsymbol{q}}_{\boldsymbol{ti}} + \boldsymbol{\eta_{pij}}(\boldsymbol{q_{ti}}) + \boldsymbol{K_{pij}}\boldsymbol{q_{ti}} + \boldsymbol{D_{pij}}\dot{\boldsymbol{q}}_{\boldsymbol{ti}} + \boldsymbol{J}_{\boldsymbol{ij}}^{\boldsymbol{T}}\boldsymbol{f_{bxi}}, \tag{26}$$

for $j = 1 \ldots n_p, i = 1 \ldots n_a$. In accordance with conditions (25) and regarding relation (15) the difference between acting torques/forces of the tree-structure and the compact parallel manipulator have to be calculated:

$$\Delta\boldsymbol{\tau_a} = \boldsymbol{\tau_c} - \boldsymbol{\tau_a} = \left(\frac{\partial \boldsymbol{q_p}}{\partial \boldsymbol{q}_{\boldsymbol{a}}^{\boldsymbol{T}}}\right)^T \boldsymbol{\tau_p}. \tag{27}$$

Since the influence of the torques/forces $\boldsymbol{\tau_e}$ of the elastic DOF on the manipulator's movement has already been taken into account during the calculation of the virtual torques $\boldsymbol{\tau_p}$, only the virtual torques (26) of the passive joints in the robot's structure are used for the calculation of these differences $\Delta\boldsymbol{\tau_a}$. Now, the new constraints torques/forces can be calculated:

$$\hat{\boldsymbol{f}}_{\boldsymbol{bxi}} = \boldsymbol{J}_{\boldsymbol{i}}^{-\boldsymbol{T}}\left[\Delta\boldsymbol{\tau_{ai}} - \boldsymbol{\tau_{pi}}\right], \tag{28}$$

for $i = 1 \ldots n_a$. The distribution of this force (28) in the manipulator's structure implies the compliance of the condition (25). Now, the external forces acting on the end-effector of the compact parallel manipulator have to be distributed between all the separate serial kinematic chains. The relations (16)–(18) can be combined:

$$\boldsymbol{f_{bx}} = \boldsymbol{US}^{+}\boldsymbol{f_x}, \tag{29}$$

with $\boldsymbol{f}_{\boldsymbol{bx}}^{\boldsymbol{T}} = \left[\boldsymbol{f}_{\boldsymbol{bx1}}^{\boldsymbol{T}} \ldots \boldsymbol{f}_{\boldsymbol{bxi}}^{\boldsymbol{T}} \ldots \boldsymbol{f}_{\boldsymbol{bxn_a}}^{\boldsymbol{T}}\right]$. These forces have to be complemented with the resulting constraint forces (28) and the force takes the end form:

$$\boldsymbol{f_{bxi}} = \boldsymbol{f_{bxi}} + \hat{\boldsymbol{f}}_{\boldsymbol{bxi}}, \tag{30}$$

for $i = 1 \ldots n_a$. The forces and torques calculated in (30) and (25) result in a force/-torque distribution within the reduced tree structure identical to the one within the original parallel manipulator. This results in identical movements of the original and the reduced tree structure.

3.5 *Verification of the Simultaneous Calculation of the Direct Dynamics*

In the new method, the Simultaneous Calculation of the Direct Dynamics (SCDD), the system is divided into a tree structure in order to accelerate the inversion of the inertia matrix. The most frequently used method, the LU-Gaussian elimination, has the complexity $O(n^3)$. For the computation of the direct dynamics in the joint space of large scale systems, it is therefore better to process several small matrices rather than a single large one. Additionally, the computations of the direct dynamics with this decomposition can be performed in parallel. In Table 1 the complexity of the matrix inversion indicated by the number of the necessary arithmetical operations is shown for three different robots of the SFB 562. These calculations were carried out with the assumption that in each kinematic chain one elastic DOF n_{ek} exists.

Table 1. Complexity of the direct dynamics calculation

	n_a	n_{pk}	n_{ek}	SCDD	L-D'A	Reduction
FIVE-BAR	2	1	1	54	64	16 %
HEXA	6	2	1	384	1728	78 %
TRIGLIDE	3	2	1	162	729	78 %

The results indicate a considerable reduction of the calculation complexity by using the proposed algorithm. Therefore each kinematic chain can be modeled with more parameters and accordingly with higher accuracy.

The new SCDD method has been compared with the L-D'A method in the simulation. In order to do this, a model of elastic planar parallel manipulators FIVE-BAR was derived [24]. The discrete elasticities k_{14} and k_{24} was considered in the upper arms of the manipulator, shown in Fig. 2a.

Q_{11} and Q_{21} represent the motors. The other parameters of this model are not relevant for this comparison. A straight line trajectory between two points p^A and p^E has been chosen. The models have then been controlled by torques created by a rigid body model without control. The black line represents the reference trajectory. The dark gray line is a result of the L-D'A model and the light gray line from the SCDD model. These results show that the models both followed the trajectory with comparable accuracy. The existing differences between the paths traveled by these two elastic models can be accounted for by the numerical precision. Figure 2b shows the distance between the endpoints of both kinematic chains. The black line $\Delta s = 1 \times 10^{-14}\, m$ shows the L-D'A and the light gray the SCDD model. The error depends on the sample interval of the simulation: the smaller the interval, the smaller the error. This error does not constitute an obstacle for the implementation of this new method because, in the field of numerical methods, algorithms are known dealing with the stabilization of the numerical calculation and the increase the computation accuracy [34].

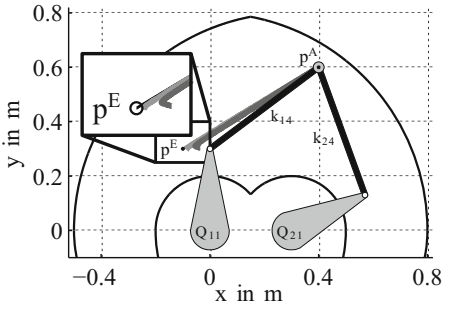


Fig. 2a. Trajectory and workspace of elastic planar parallel manipulator - FIVE-BAR

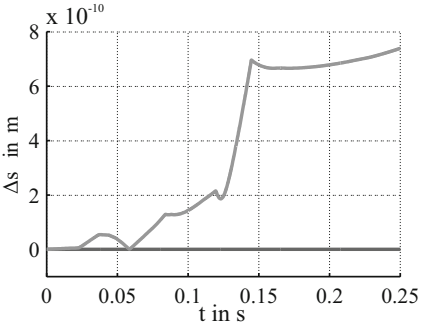


Fig. 2b. Distance between the two serial kinematic chains

4 Control of the Structural Vibrations of the Parallel Manipulator

One of the principal requirements for the structural as well as the motion control is a constant control quality all over the workspace. The properties of the manipulator are pose-dependent; this implies the necessity of the use of a model based control algorithm [5, 35]. Furthermore the algorithm has to be robust enough to be able to tolerate uncertainties of the robot model. According to the singular perturbation method, the control law for vibration suppression of the robot structure can be designed separately from the motion control of the manipulator [2, 3].

In this chapter a model-based control algorithm is described in Sect. 4.1 along with the update strategy in Sect. 4.2 to adjust the controller parameters to the varying nonlinear plant. The approach is validated in Sects. 4.3 and 4.4 using the experimental results obtained with the FIVE-BAR manipulator.

4.1 Model-Based Linear Controller

The choice of the control structure was motivated by methods for passive vibration suppression, which are commonly applied to linear systems [36]. These methods use additional spring-mass-damper-systems as dynamic vibration absorbers, which are installed with poorly damped systems and tuned to suppress their natural frequency. The main parameters and the diagram of a single mode model used for the derivation of the controller structure is displayed in Fig. 3a. In the selected approach the actuator is chosen to behave like a spring resulting in a controller with a D_2T_2 transfer function:

$$G_a(s) = \frac{\mathcal{L}\{\ddot{x}_a\}}{\mathcal{L}\{f_s\}} = \frac{v_a s^2}{m_{mod} s^2 + d_a s + k_a}\,, \tag{31}$$

with d_a denoting the virtual attenuation factor, k_a the virtual stiffness, x_a the actuating variable and v_a the controller gain. Since the structure exhibits only a very

low damping of the fundamental modes, their damping factor can be neglected ($d_{mod} \approx 0$) [37]. The effect of the controller upon the system mainly equals the product of the accelerations produced by the actuator $\ddot{x}_a$ and the effective vibrating mass m_{mod}. Therefore, the transfer function $G_a(s)$ describes the relationship between the force measured by the sensor f_s and the mentioned acceleration $\ddot{x}_a$.

The aim of the controller design is to choose the parameters of the virtual spring-mass-damper-system formed by the actuator in a way that allows the closed-loop system to return to its resting point as quickly as possible without oscillating. Therefore the closed-loop system has to show the behavior of a critically damped second order system. By comparing the coefficients of the closed-loop system with the desired transfer function the parameters k_a and d_a can be determined. The virtual stiffness k_a of the actuator is:

$$k_a = \frac{4}{\pi^2 - 4} k_{mod} = \beta_{mod} k_{mod}, \tag{32}$$

with β_{mod} the factor between both stiffnesses. The virtual attenuation factor has to ensure very good damping of the system and is derived as follows:

$$d_a = 2\sqrt{k_a m_{mod}}. \tag{33}$$

The parameters k_{mod} and m_{mod} represent the stiffness and the effective vibrating mass of a single structural mode projected into the direction of the actuator.

A modal controller is designed for each controlled structural mode of the manipulator and each of the used actuators. The actuator response is computed by superposition of the modal controllers associated with it (see Fig. 3b). Further information regarding the used formula and the proof of the robustness and the stability of the selected controller scheme are detailed in [24]. Since most

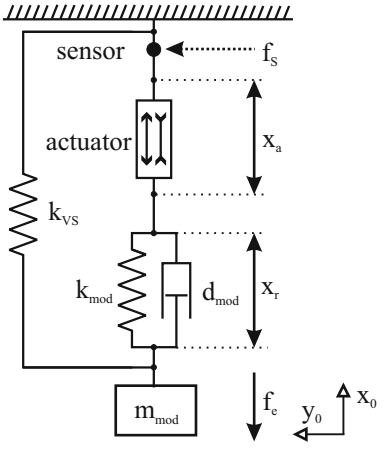


Fig. 3a. Single mode model

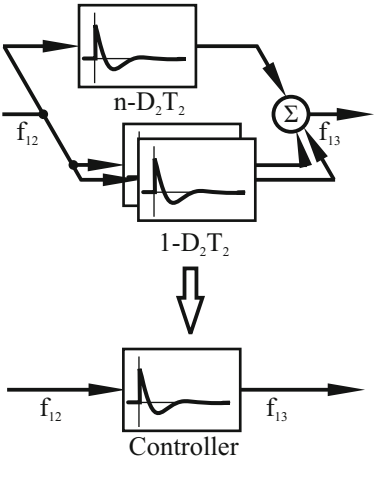


Fig. 3b. Superposition of the modal controllers.

manipulators belong to the family of nonlinear systems, the frequencies of their respective modes change during motion. Consequently, the parameters k_{mod} and m_{mod} have to be updated according to the end-effector position.

4.2 Pose-Dependent Adjustment of the Controller Parameters

A linear controller was proposed for the control of the structural vibrations. In order to adjust the modal controller to the varying plant the features of the force and inertia ellipsoids were utilized [16, 21].

The force ellipsoids allow the determination of the robot properties with regard to the generation of forces and the resistance against forces and, therefore, also the calculation of the structural stiffness in an arbitrary direction in space. The stiffness can be derived as follows:

$$k_{mod} = \sqrt{\left(\boldsymbol{f}_x^T \boldsymbol{N}_x^T \boldsymbol{N}_x \boldsymbol{f}_x\right)^{-1}} \tag{34}$$

with $\boldsymbol{N}_x = \boldsymbol{K}_x^{-1}$ denoting the compliance matrix (the inverse of the stiffness matrix in the task coordinate frame), the unit vector $\|\boldsymbol{f}_x\| = 1$ describing the spatial direction. In some cases it is advisable to generate the fields for each mode to be able to analyze them separately.

The inertial ellipsoid describes the ability of the manipulator to generate a unit force in an arbitrary spatial direction; it also determines the inertia of the robot structure into a given direction. The effective vibrating mass differs from the projected inertia by a factor λ and can be calculated from:

$$m_{mod} = \lambda \sqrt{\ddot{\boldsymbol{x}}_x^T \boldsymbol{M}_x^T \boldsymbol{M}_x \ddot{\boldsymbol{x}}_x} \tag{35}$$

with the unit vector $\|\ddot{\boldsymbol{x}}_x\| = 1$ determining the spatial direction and $\boldsymbol{M}_x^T$ describing the inertia matrix of the manipulator in the task frame [24]. Here again, the analysis of separated modes may require the separate calculation of each field.

The compliance matrix $\boldsymbol{N}_x$ and the inertia matrix $\boldsymbol{M}_x$ are pose-dependent and have to be calculated in each control step using the nonlinear model of the manipulators dynamics, as derived in Sec. 3. Thus the control algorithm consists of the transfer function (31), the parameters of the controller (32) and (33) and the adaption formulas (34) and (35).

4.3 Verification of the Proposed Control Law

The proposed control algorithm has been tested with the elastic planar parallel manipulator FIVE-BAR (see Fig. 4a). In the robot model one lumped elasticity was included for each upper arm of the manipulator, depicted in Fig. 4b as k_{14} and k_{24}. The parameters k_{14}, m_{12} and k_{24}, m_{22} model the properties of the section of the piezoelectric stack used as a force sensor.

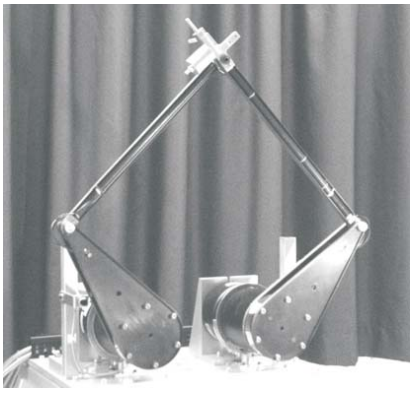


Fig. 4a. Elastic planar parallel robot FIVE-BAR

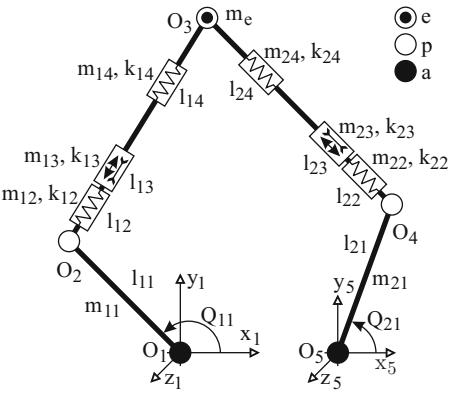


Fig. 4b. Model's parameter of FIVE-BAR

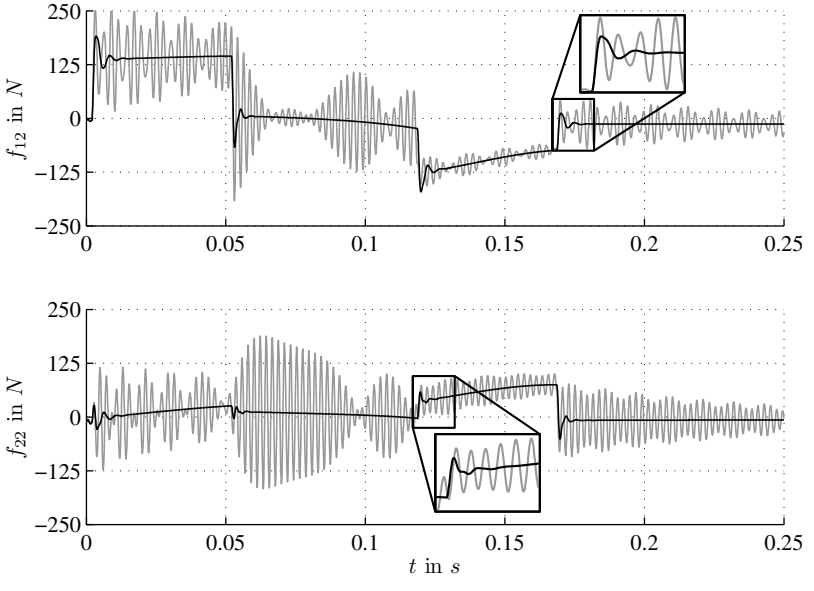


Fig. 5. Vibration suppression on the elastic parallel manipulator FIVE-BAR

The controller was activated during a motion along a straight line trajectory, as shown in Fig. 2a. Every actuator was driven by a single mode controller. The force measured at the respective piezo sensor served as an input for the controller.

Figure 5 depicts the forces f_{12} and f_{22} at the sensors during the motion along the given trajectory with (black) and without (light gray) structural control. The

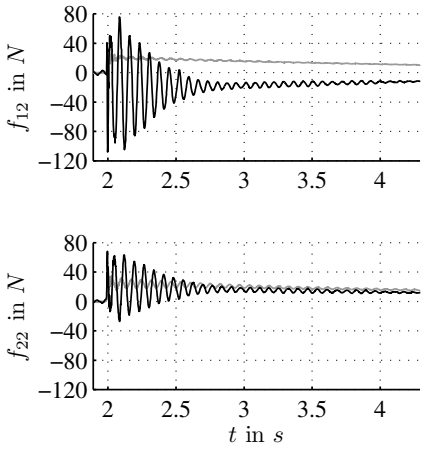


Fig. 6a. Measurement using piezo sensors

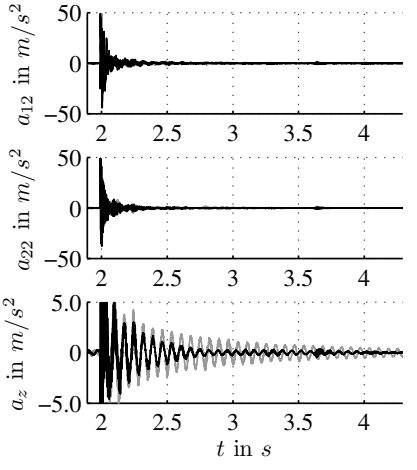


Fig. 6b. Measurement using a 3-axial accelerometer

vibrations of the elastic modes were excited by the stepwise change of the accelerations caused by the drive moments. Without the controller, distinct vibrations are visible, whereas they are efficiently attenuated after the activation of the control, during the motion as well as after the completion of the trajectory. The magnifications in Fig. 5 provide further details for two points on the trajectory. The time scale is stretched by 1.5 and 2.5, respectively. The effort of the actuators and thus the control quality remains nearly constant in each point of the trajectory.

4.4 Experimental Results

The presented experiments were carried out for one pose of the manipulator ($Q_{11} = 135°$ and $Q_{12} = 45°$) and with an additional payload at the end-effector $m_e = 2\,kg$. The dominant oscillations were directed into the z-direction, perpendicular to the working plane. Therefore only measurements of accelerations into the z-direction, obtained with the acceleration sensor attached to the end-effector, were used for the control strategy. The results of the controller action are shown in Fig. 6a for the piezo sensors and in Fig. 6b for the 3-axial acceleration sensor. The attenuation of the z-components of oscillations at the end-effector was $7.3\,dB$. These experimental results have confirmed the applicability of the new control algorithm and the chosen design strategy.[1] Some implementation difficulties, e.g. the phase shift due to the digital signal processing, still need to be addressed in future work in order to apply the proposed controller scheme to a wider range of mechanisms.

[1] A detailed discussion of the experimental and simulative results can be found in [24].

5 Conclusions

In this contribution the modeling and control of structural vibrations of elastic parallel manipulators has been discussed. For the modeling of the manipulators' dynamics, an enhancement of the Lagrange-D'Alembert formulation has been proposed using a new algorithm for the derivation of the Jacobian of parallel manipulators. Based upon these results, a method has been presented to simultaneously calculate the direct dynamics of parallel and furthermore elastic parallel manipulators. It allows a significant reduction of the complexity of the calculations. This procedure has been demonstrated for the FIVE-BAR manipulator. The knowledge of manipulator's dynamics has allowed the design of a new model-based control law for vibration suppression in the manipulator's structure. A robust linear controller combined with a pose-dependent adjustment law has been conceptualized and presented. The performance and potential of the new algorithm has been confirmed by simulative and experimental examinations. Future work will be directed towards applying the presented methods to parallel kinematic machines offering more degrees of freedom.

References

1. Keimer, R., Sinapius, M.: Design and Implementation of Adaptronic Robot Components. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 413–427. Springer, Heidelberg (2010)
2. Hermle, M.: Hierarchische Regelung globaler Bewegungen elastischer Roboter. Fortschritt-Berichte, vol. 8. VDI, Stuttgart (2001)
3. Siciliano, B., Book, W.J.: A singular perturbation approach to control of lightweight flexible manipulators. 7(4), 79–90 (1988)
4. De Luca, A., Panzieri, S., Ulivi, G.: Stable inversion control for flexible link manipulators. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp. 799–805. IEEE, Leuven (1998)
5. Moallem, M., Patel, R.V., Khorasani, K.: Flexible-link robot manipulators: Control techniques and structural design. Lecture Notes in Control and Information Sciences, vol. 257. Springer, London (2000)
6. Yang, H., Krishnan, H., Ang, M.H.: Tip-trajectory tracking control of single-link flexible robots by output redefinition. In: Proc. of IEE Control Theory and Applications, vol. 147, pp. 580–587. IEEE, Los Alamitos (2000)
7. Wierach, P., Schönecker, A.: Bauweisen und Anwendungen von Piezokompositen in der Adaptronik. In: Adaptronic Congress, Göttingen, pp. 1–11 (2005)
8. Gabbert, U., Koeppe, H., Nestorovic-Trajkov, T.: Entwurf intelligenter Strukturen unter Einbeziehung der Regelung. In: Automatisierungstechnik, vol. 50, pp. 432–438. Oldenbourg Verlag, München (2002)
9. Rose, M.: Modal based correction methods for the placement of piezoceramic modules. In: Proc. of ASME International Mechanical Engineering Congress and Exposition, Orlando, Florida, USA (2005)
10. Algermissen, S., Rose, M., Keimer, R., Breitbach, E.: High-speed parallel robots with integrated vibration-suppression for handling and assembly. In: Proc. of the 11th Annual International Symposium on Smart Structures and Materials, pp. 1–10. SPIE, San Diego (2004)

11. Algermissen, S., Rose, M., Sinapius, M., Stachera, K.: Robust gain-scheduling control for parallel robots with smart-structure components. In: Schuetz, D., Raatz, A., Wahl, F.M. (eds.) Proc. of the 3rd International Colloquium of the Collaborative Research Center SFB 562 - Robotic Systems for Handling and Assembly. Fortschritte in der Robotik, vol. 14, pp. 191–206. Shaker, Braunschweig (2008)
12. Breitbach, E., Algermissen, S., Keimer, R., Rose, M., Stachera, K.: Adaptive Tools in Parallel Robotics. In: Last, P., Budde, C., Wahl, F.M. (eds.) Proc. of the 2nd International Colloquium of the Collaborative Research Center 562 - Robotic Systems for Handling and Assembly, p. 203. Shaker, Aachen (2005)
13. Wang, X., Mills, J.K.: Substructuring dynamic modeling and active vibration control of a smart parallel platform. In: Proc. of 2004 ASME International Mechanical Engineering Congress (IMECE 2004), pp. 1–8. ASME, Anaheim (2004)
14. Algermissen, S., Sinapius, M.: Robust gain scheduling for smart structures in parallel robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 159–174. Springer, Heidelberg (2010)
15. Murray, R.M., Li, Z., Sastry, S.S.: A mathematical introduction to robotic manipulation, 1st edn., p. 456. CRC Press LLC, Boca Raton (1994)
16. Tsai, L.-W.: Robot analysis: the mechanics of serial and parallel manipulators. John Wiley and Sons, Inc., Chichester (1999)
17. Kock, S.: Parallelroboter mit Antriebredundanz. Fortschritt - Berichte, vol. 8. VDI, Düsseldorf (2001)
18. Stachera, K.: A new method for the direct dynamics calculation of parallel manipulators. In: Proc. of the 6th World Congress on Intelligent Control Automation (WCICA), Dalian, China. IEEE, Los Alamitos (2006)
19. Khalil, W., Gautier, M.: Modelling of mechanical system with lumped elasticity. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp. 3965–3970. IEEE, San Francisco (2000)
20. Robinett, R.D., Dohrmann, C., Eisler, G.R., Feddema, J., Parker, G.G., Wilson, D.G., Stokes, D.: Flexible robot dynamics and controls. Kluwer Academic/Plenum Publishers, International Federation for System Research - IFSR, New York (2002)
21. Nakamura, Y., Ghodoussi, M.: Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators, vol. 5(3), pp. 294–302 (1989)
22. Park, F.C., Choi, J., Ploen, S.R.: Symbolic formulation of closed chain dynamics in independent coordinates. Pergamon: Mechanism and Machine Theory 34, 731–751 (1999)
23. Stachera, K., Schumacher, W.: Simultaneous calculation of the direct dynamics of the elastic parallel manipulators. In: Proc. of the 13th IEEE IFAC International Conference on Methods and Models in Automation and Robotics (MMAR). IEEE IFAC, Szczecin (2007)
24. Stachera, K.: Konzepte für elastische, parallele Manipulatoren zur Regelung der Strukturschwingungen. Ph.D. thesis, Institute of Control Engineering, TU Braunschweig, Germany (2009)
25. Czichos, H. (ed.): HÜTTE, Die Grundlagen der Ingenieurwissenschaften, 29th edn. Springer, Heidelberg (1989)
26. Gross, D., Hauger, W., Wriggers, P.: Technische Mechanik 4, 6th edn. Springer, Heidelberg (2007)
27. Benedict, C.E., Tesar, D.: Model formulation of complex mechanisms with multiple inputs: Part II - The dynamic model. Journal of Mechanical Design 100, 755–761 (1978)
28. Isermann, R.: Mechatronische Systeme: Grundlagen, 1st edn. korrigierter Nachdruck. Springer, Heidelberg (2002)

29. Merlet, J.-P.: Parallel Robots. Kluwer Academic Publishers, Dordrecht (2000)
30. Stachera, K.: An approach to direct kinematics of a planar parallel elastic manipulator and analysis for the proper definition of its workspace. In: Proc. of the 11th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR). IEEE, Miedzyzdroje (2005)
31. Wang, X., Mills, J.K.: A FEM model for active vibration control of flexible linkages. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp. 4308–4313. IEEE, New Orleans (2004)
32. Stachera, K., Schumacher, W.: Derivation and calculation of the dynamics of elastic parallel manipulators. In: Automation and Robotics, 1st edn, pp. 261–276. I-Tech Education and Publishing, Vienna (2008) ISBN 978-3-902613-41-7
33. Miller, K., Clavel, R.: The Lagrange-based model of Delta-4 robot dynamics. In: Robotersysteme, vol. 8, pp. 49–54. Springer Verlag, Germany (1992)
34. Baumgarte, J.: Stabilization of constraints and integrals of motion in dynamical systems. Computer Methods in Applied Mechanics 1, 1–36 (1972)
35. Spong, M.W., Vidyasagar, M.: Robot dynamics and control. John Wiley and Sons, Inc., Chichester (1989)
36. Fischer, U., Stephan, W.: Mechanische Schwingungen. Fachbuchverlag Leipzig - Köln (1993)
37. Stachera, K., Schumacher, W.: Robust vibration control of flexible planar parallel robot. In: Proc. of the 9^{th} IEEE International Conference on Methods and Models in Automation and Robotics (MMAR). IEEE, Miedzyzdroje (2003)

Redundant Parallel Kinematic Structures and Their Control

Sönke Kock and Walter Schumacher

Abstract. The design and experimental verification of a control system for fast parallel robots with actuation redundancy was pioneered with the prototype PA-R-MA. It is shown how actuation redundancy can improve the capabilities of parallel robots, and how a control can be implemented. A particular focus has been put on verifying selected redundancy strategies that are well-known from literature, but whose applicability for fast parallel robots had not yet been verified, in particular in conjunction with designs that use gear transmissions rather than gearless servo motors. Undesired stress peaks caused by the inherently overconstrained structure were reduced by a factor of 6. To achieve this, a measurement and feedback control of structural forces was developed and verified. As a result, we can show that the actuation redundant manipulator can be moved at very high accelerations without losing control of the internal stress levels. One investigated objective is anti-backlash control, which resulted in an increase of positioning repeatability by a factor of 7.

1 Introduction and Prior Work

Actuation redundancy occurs whenever a mechanism has more actuated joints than internal kinematic degrees of freedom (d.o.f.). This happens only if at least one closed kinematic chain is formed, be it during part of the task (stiff contact of an open chain with the environment) or by design (closed-loop mechanisms). The redundancy creates additional constraints and thus internal force dimensions ("force null space"). If the null space can be controlled, force or torque distribution among

Sönke Kock
ABB Corporate Research, Västerås, Sweden
e-mail: soenke.kock@se.abb.com

Walter Schumacher
Technische Universität Braunschweig, Institute of Control Engineering,
Hans-Sommer-Straße 66, 38106 Braunschweig, Germany
e-mail: w.schumacher@tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 143–157.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

Fig. 1. Parallel Redundant Manipulator Prototype

the actuators can be optimized by some (minimal 2-norm or ∞-norm [1], [2], active stiffness [3], [4], anti-backlash-control [5] etc.).

Actuation redundancy is not to be confused with kinematic redundancy known from serial-type robots, when extra actuated joints add kinematic d.o.f. and hence a "motion null space". For a detailed introduction to force and motion redundancy and their duality, the reader is referred to [6] and [7].

Actuation redundancy has been investigated for various alternative mechanisms such as dexterous fingers [8], robot wrists [9] and shoulders [4], walking machines [10], [11] and haptic displays [1]. It came as a feasible solution for increasing payload and acceleration of directly driven robots with limited torques [12]. All previous control approaches known to the authors have in common that regulation of internal forces is done in a feed-forward fashion, using computed motor current references. This is usually facilitated by the use of direct or semi-direct drives, where low gear reductions give good back drivability and hence less tendency for the appearance of large structural forces.

In this article, we show that by adding a force control loop for the internal forces, redundancy control schemes can be used with high accuracy even at comparably high speeds and accelerations. The authors believe that the prototype PA-R-MA presented in this article is the fastest robot with actuation redundancy to date.

2 PA-R-MA Prototype and Kinematic Modeling

The PA-R-MA (Parallel Redundant Manipulator) was developed as a test bed for investigating control issues with actuation redundancy.

The servo driven manipulator is equipped with torque sensors at the output of the gearboxes. The sensors are strain gauges that are applied to the crank levers and that pick up link deflections. In a mechatronic design approach, the stiffness of the robot structure was designed to match the required sensing range and strain for maximum sensing quality. Yet the stiffness of the manipulator matches the stiffness of the gearboxes well, so no unacceptable compromise in the overall manipulator stiffness had to be made by introducing torque sensing.

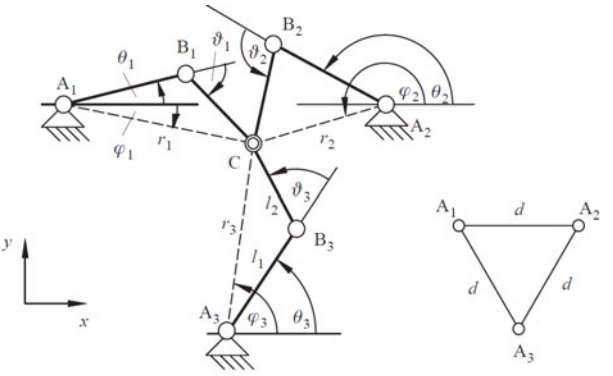


Fig. 2. Coordinates and kinematic parameters

2.1 Coordinates and Geometric Parameters

PA-R-MA comprises three actuated legs A_iB_iC that meet at a center joint C, where the tool is mounted. Fig. 2 shows the joint coordinates and kinematic parameters as well as some dependant coordinates used for the inverse kinematics calculation.

2.2 Forward Kinematics

The position of the center joint C is described by the generalized coordinates (x,y) – a rotational d.o.f. is not defined. Instead, a third coordinate will later be introduced that defines the null space torque or amount of pre-stress of the manipulator structure. The three actuated joints A_1 to A_3 comprise geared servo motors with motor-side position sensors. The usage of 3 sensors for 2 motion d.o.f. makes sensor redundancy inherent – it leads to the advantage that a forward kinematic solution can be derived that is unambiguous and globally valid throughout the workspace [13]:

$$x = \frac{|x_{B1}|^2 (y_{B2} - y_{B3}) + |x_{B2}|^2 (y_{B3} - y_{B1}) + |x_{B3}|^2 (y_{B1} - y_{B2})}{2 (x_{B1} (y_{B2} - y_{B3}) + x_{B2} (y_{B3} - y_{B1}) + x_{B3} (y_{B1} - y_{B2}))} \tag{1}$$

$$y = \frac{|x_{B1}|^2 (x_{B3} - x_{B2}) + |x_{B2}|^2 (x_{B1} - x_{B3}) + |x_{B3}|^2 (x_{B2} - x_{B1})}{2 (x_{B1} (y_{B2} - y_{B3}) + x_{B2} (y_{B3} - y_{B1}) + x_{B3} (y_{B1} - y_{B2}))} \tag{2}$$

where

$$\begin{bmatrix} x_{Bi} \\ y_{Bi} \end{bmatrix} = \begin{bmatrix} x_{Ai} \\ y_{Ai} \end{bmatrix} + l_1 \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix}. \tag{3}$$

It is confirmed in [14], [15] that the accuracy of this solution under the assumption of sensor noise and other uncertainties is favorable, and can only be improved marginally with relatively high computational effort by employing statistical or numerical methods.

2.3 Inverse Kinematics

For the inverse kinematics, we obtain from any triangle A_iB_iC in Fig. 1 the inverse kinematic relation (see [13] for details):

$$\theta_{1,2} = \pm \arccos\left(\frac{l_1^2 - l_2^2 + (x - x_{\mathrm{A}i})^2 + (y - y_{\mathrm{A}i})^2}{2l_1\sqrt{(x - x_{\mathrm{A}i})^2 + (y - y_{\mathrm{A}i})^2}}\right) + \arctan\left(\frac{x - x_{\mathrm{A}i}}{y - y_{\mathrm{A}i}}\right) \tag{4}$$

Two solutions are originally obtained because of the uncertainty of the arccos function, indicated by the ± sign. They correspond to different assembly modes of the legs (elbow up, elbow down). By mechanical limits of joints B_i it is prevented that the leg configurations can change – the inverse kinematics therefore becomes unambiguous. By choosing a regular configuration as depicted in Fig. 1, which gives a more uniform dexterity distribution [13], all three inverse kinematic equations are identical except that index i varies.

3 Modeling and Decoupling

In this chapter, we describe how the manipulator can be modeled for control purposes, and how the null space is decoupled from the motion space to allow for decoupled redundancy control.

3.1 Statics and Jacobian

The forward static relationship between input torques and output force is defined by a linear Jacobian mapping

$$\boldsymbol{f} = \boldsymbol{G}\boldsymbol{T} \tag{5}$$

where $\boldsymbol{f}$ is the $n \times 1$-dimensional vector of output forces (n being equivalent to the motion degrees of freedom of the robot), $\boldsymbol{T}$ is the $m \times 1$-dimensional vector of input torques (m being the number of independent actuators), and where $\boldsymbol{G}$ is the $n \times m$-dimensional Jacobian matrix. Fig. 3 shows a graphical interpretation of the vector spaces $\boldsymbol{f}$ and $\boldsymbol{T}$. The forward static transformation of any torque vector $\boldsymbol{T} = [\tau_1\, \tau_2\, \tau_3]^{\mathrm{T}}$ is unambiguous – however, the inverse is not. The 2-dimensional subspace M which contains the torque vectors with minimum 2-norm is defined by the inverse static relationship

$$\boldsymbol{T} = \boldsymbol{G}^{+}\boldsymbol{f}, \tag{6}$$

using the generalized matrix inverse

$$\boldsymbol{G}^{+} = \boldsymbol{G}^{\mathrm{T}}\left(\boldsymbol{G}\boldsymbol{G}^{\mathrm{T}}\right)^{-1}. \tag{7}$$

However, an infinite number of other solutions exist. They contain components of the torque vector perpendicular to M, which is the vector space Ker($\boldsymbol{G}$). The sets of motor torques lying in the null space are called antagonistic torques as they act against

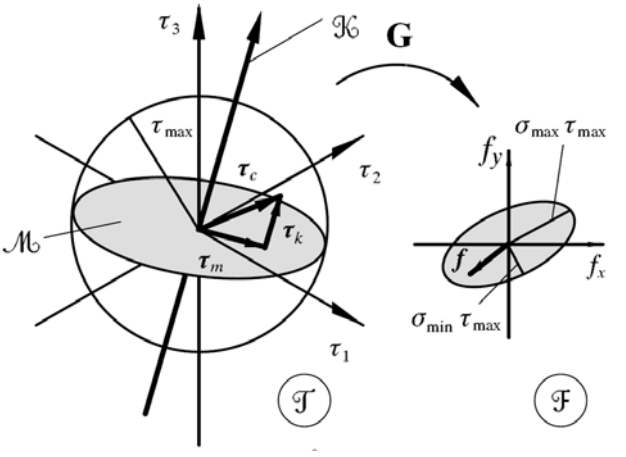


Fig. 3. 2×3 Jacobian mapping and geometrical interpretation

each other such that they create structural forces, but no movement. In the case of the PA-R-MA robot with one-dimensional null space, we define the unity vector $\boldsymbol{\tau}_{k,0}$, which is calculated by the cross product of the two columns of $\boldsymbol{G}^{+}$. The projection and normalization of any torque vector $\boldsymbol{\tau}$ onto the null space is calculated by

$$\boldsymbol{\tau}_{k,0} = \frac{(\boldsymbol{I} - \boldsymbol{G}^{+}\boldsymbol{G})\,\boldsymbol{\tau}}{\|(\boldsymbol{I} - \boldsymbol{G}^{+}\boldsymbol{G})\,\boldsymbol{\tau}\|}. \tag{8}$$

The inverse static solution with null space component can then be decomposed as

$$\boldsymbol{T} = \boldsymbol{G}^{+}\boldsymbol{f} + \boldsymbol{\beta}\boldsymbol{\tau}_{k,0}, \tag{9}$$

where $\boldsymbol{\beta}$ is the scaling factor of the vector of null space torques. It is a $(m-n \times m-n)$-sized diagonal matrix. When the null space is one-dimensional as it is the case with PA-R-MA, β is a scalar. Hence, the static motor torques can be selected to balance an external force $\boldsymbol{f}$ plus a null space component that can be scaled arbitrarily within the available torque range.

The null space coordinate β of an arbitrary torque vector can be calculated by calculating the inner product

$$\beta = \boldsymbol{\tau}_{k,0}^{T}\boldsymbol{\tau}. \tag{10}$$

Interestingly, in the case of multi-dimensional null spaces (i.e. by adding additional actuators), the robot structure can be pre-stressed in arbitrary directions by scaling the $m - n$ unit vectors of the null space basis appropriately.

3.2 Dynamic Modeling and Decoupling

Since two motion d.o.f. (x, y) and one pre-stress degree of freedom (β) exist, it is straightforward to assign $[x, y, \beta]$ as the generalized coordinates of the PA-R-MA.

This is particularly useful since unambiguity is ensured for both the forward and the inverse kinematics, which means that at any time a one-to-one relationship between end effector position and joint coordinates exists. It is therefore straightforward to also set up a decoupled control system for the generalized coordinates, which later leads to a Cartesian motion controller and a decoupled null space controller for the pre-stress.

3.3 Equations of Motion

The motion rigid-body dynamic model can be found in [16]. It is also shown that drastic simplifications can be made by assuming a point mass m_C at joint C and weightless links B_iC_i with good accuracy. The links are designed from light weight aluminum, as they experience only push pull forces and virtually no moments, and this is why their effect on dynamics is low and can be neglected.

The sum of Cartesian inertial and external forces acting on joint C are then mapped onto joint torque level by the inverse static relationship (6), so that the equation of motion on joint level becomes

$$\boldsymbol{\Theta}\ddot{\boldsymbol{\theta}} + \boldsymbol{k}_v\dot{\boldsymbol{\theta}} + \boldsymbol{G}^{+}m_C\ddot{\boldsymbol{x}} = \boldsymbol{T}, \tag{11}$$

where $\boldsymbol{T}$ may still contain null space torques that don't affect the acceleration as they cancel out when transformed to Cartesian space, $\boldsymbol{\Theta}$ is a 3×3 diagonal matrix containing the lumped inertia of the drive trains including the crank shaft A_iB_i and joint B_i, $\boldsymbol{k}_v$ describes the speed dependant friction. The model could be extended by adding Coulomb friction or forces acting on the manipulator [17]. By multiplying with $\boldsymbol{G}$ from the left,

$$\boldsymbol{G}\boldsymbol{\Theta}\ddot{\boldsymbol{\theta}} + \boldsymbol{G}\boldsymbol{k}_v\dot{\boldsymbol{\theta}} + \boldsymbol{G}\boldsymbol{G}^{+}m_C\ddot{\boldsymbol{x}} = \boldsymbol{G}\boldsymbol{T}, \tag{12}$$

and by using the inverse velocity and acceleration relationships $\dot{\boldsymbol{\theta}} = \boldsymbol{G}^{\mathrm{T}}\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{\theta}} = \boldsymbol{G}^{\mathrm{T}}\ddot{\boldsymbol{x}} + \dot{\boldsymbol{G}}^{\mathrm{T}}\dot{\boldsymbol{x}}$, the Cartesian equations of motion are obtained:

$$\boldsymbol{M}_x(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{C}_x(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} = \boldsymbol{G}\boldsymbol{T}, \tag{13}$$

with the Cartesian mass inertia matrix

$$\boldsymbol{M}_x(\boldsymbol{x}) = \boldsymbol{\Theta}\boldsymbol{G}\boldsymbol{G}^{\mathrm{T}} + \begin{bmatrix} m_C & \\ & m_C \end{bmatrix}, \tag{14}$$

and the Cartesian matrix of centrifugal and Coriolis forces

$$\boldsymbol{C}_x(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \boldsymbol{k}_v\boldsymbol{G}\boldsymbol{G}^{\mathrm{T}} + \boldsymbol{\Theta}\boldsymbol{G}\dot{\boldsymbol{G}}^{\mathrm{T}}. \tag{15}$$

In this model, it is assumed that all drive trains are identical, i.e. have the same inertia and friction parameters. It is shown in [3] that variations in friction parameters between them create a coupling between null space and motion space. It will

be observed later that this coupling exists in reality, but is handled well by the null space controller.

3.4 Null Space Dynamics

In [17] the authors show that the null space dynamics, which corresponds to the internal inertia and impedance of the robot structure when responding to pre-stress, can be modeled as a linear system with constant parameters irrespective of the robot position. This holds under the assumptions that the longitudinal compliance of the links B_iC_i and the compliance of the joints B_i and C_i can be neglected when compared to the bending compliance of crank shafts A_iB_i. It was shown and verified by experiment that a well-damped linear second order system is accurate enough to model the null space dynamics for the purpose of controller design:

$$\beta(s) = \frac{n_g}{\frac{s^2}{\omega_0^2} + 2D\frac{s}{\omega_0} + 1}\beta_{\text{mot}}(s). \tag{16}$$

β_{mot} is the null space coordinate vector on the motor side of the actuator gearboxes, and n_g is the gearbox ratio. For the PA-R-MA manipulator, the following parameters were identified [17]:

$$\omega_0 = 138\text{s}^{-1} \tag{17}$$

$$D = 0.9. \tag{18}$$

It can be concluded from these findings that the null space dynamics is approximately linear, well damped and invariant to manipulator position which is a good precondition for linear feedback control, if the null space is properly decoupled from the motion space. It is also worth noting that since there exists a linear compliance

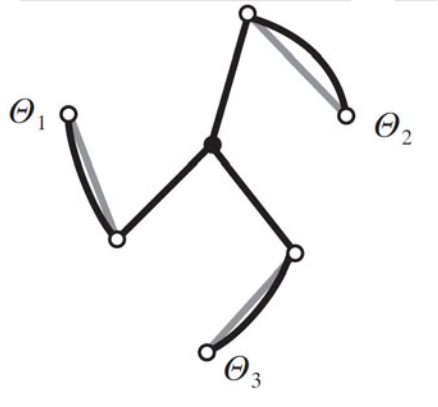


Fig. 4. Illustration of null space deflections

relationship between the strain of the crank shafts (as picked up by the strain gauges) and the corresponding material stress (which represents the actuator torque) with negligible measurement delays, equation (16) can also be expressed for null space deflections rather than null space torques. Results and control approach would be equivalent. Fig. 4 sketches how null space torques result in deflections in the drive train and crank shafts, resulting in incremental null space movements of joints A_i, but no movement of the joints B_i and C.

4 Controller Design

With the above choice of generalized coordinates and the knowledge of the system physics, the controller structure can be established.

4.1 Control Structure

Fig. 5 shows how the controller is implemented. Four computational blocks are needed to decouple the null space from the motion space (starting from the bottom):

- The forward kinematics FK (1,2) transforms joint positions into end effector space
- $\boldsymbol{G}^{+}$ transfers the Cartesian control output, which corresponds to a force, into actuator torque space, which act as torque references for the drive systems
- $\boldsymbol{\tau}_{\mathrm{k},0}$ is the base vector of the null space, which is scaled with the null space controller output. The resulting torque reference is superimposed on the motion torque references
- $\boldsymbol{\tau}_{\mathrm{k},0}^{\mathrm{T}}$ represents the inner product that projects the measured torques of the driving joints, as picked up by the strain gauges, onto the torque null space, resulting in the actual length of the null space component, β.

4.2 Cartesian Motion Controller

The motion controller as depicted in Fig. 5 controls a 2-dimensional robot system with tight dynamic couplings (13,14,15). A model-based decoupling of motion direction is first employed which does three things:

- To compensate dynamic cross couplings caused by the Cartesian mass matrix
- To compensate dynamic cross couplings and disturbances by centrifugal and Coriolis forces
- To adapt the controller gains inversely to changes in inertia changes of the main diagonal of the mass matrix.

These requirements are met by the control

$$\boldsymbol{f}_{\mathrm{ref}} = \boldsymbol{M}_x(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{C}_x(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}}_{\mathrm{ref}}, \tag{19}$$

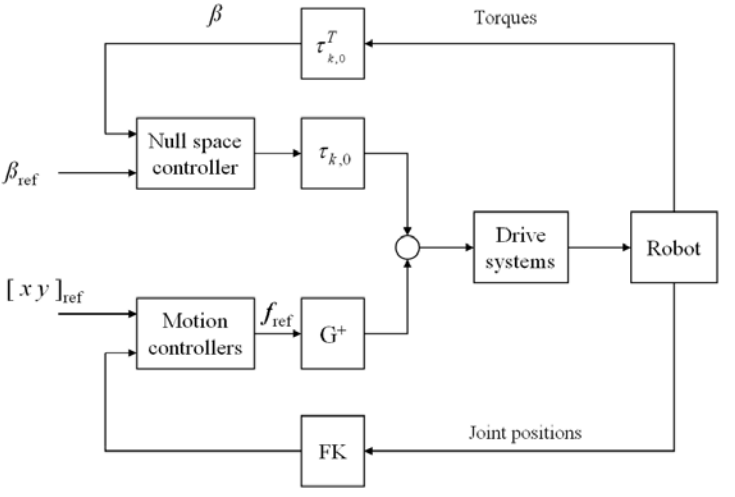


Fig. 5. Controller structure

where $\boldsymbol{u}$ is the vector of outputs of the motion controllers for the x and y directions. Due to the decoupling achieved with (19), the motion controllers can be designed as pure servo controllers, satisfying with robotic requirements:

- No overshoot
- Good disturbance rejection to assure path accuracy
- Trajectory planner that generates reference speeds and accelerations for feed-forward purposes.

Based on these requirements, the following motion controller was designed (see Fig. 6): The feedback controller is of cascaded PID structure, while the feed-forward filters compensate for time constants of the speed calculation (T_1) and of the control loop in the power amplifiers (T_2). For more details about the servo controller design, see [16].

4.3 Null Space Controller

For the null space control, a modified PID controller is used that is designed using pole placement [18]. A feed-forward factor is introduced that allows tuning the tracking behavior independent of the loop performance. The eigenfrequency of the control loop was chosen to be 140rad/s. This corresponds to the natural frequency of the null space as we know from the modeling described in chapter 3.4, i.e. the null space controller does not speed up the null space dynamics; it just regulates steady state accuracy and damping, which will later prove sufficient for dynamic motions. Fig. 7 shows the step response of the controlled null space with a pre-stress

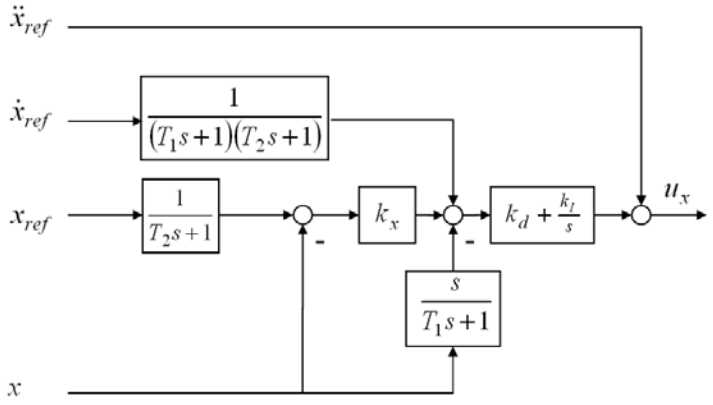


Fig. 6. Servo controller

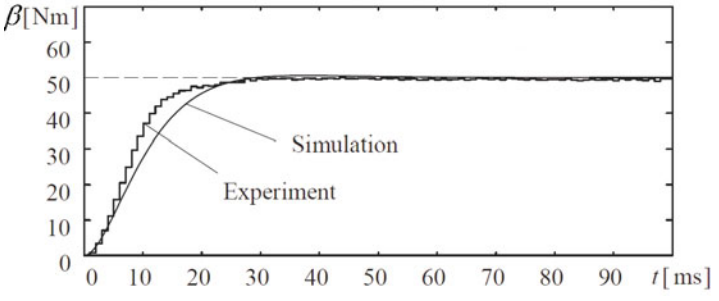


Fig. 7. Step response of pre-stress coordinate

reference change of 50Nm. The controller shows good performance: The step response is well damped and has a settling time of 30ms.

5 Experimental Verification

For reasons of compactness, no details about the control system implementation are presented in this article. The reader is referred to [16].

5.1 Dynamic Tests

The test trajectory for the dynamic tests is shown in Fig. 8: The end effector moves along the equilateral triangle of length 25cm in the order $A - B - C - A$ (see Fig. 8). The speed is reduced to zero at every corner ("fine point"), and a pause of 100ms is inserted there. The acceleration and deceleration profiles are jerk limited. Acceleration varies long the trajectory between 4 and 5g, and the linear speed between 2

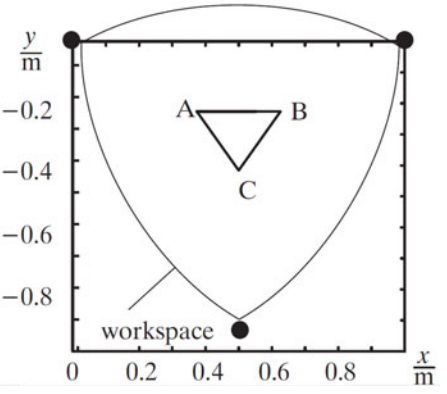


Fig. 8. Test trajectory $A - B - C - A$

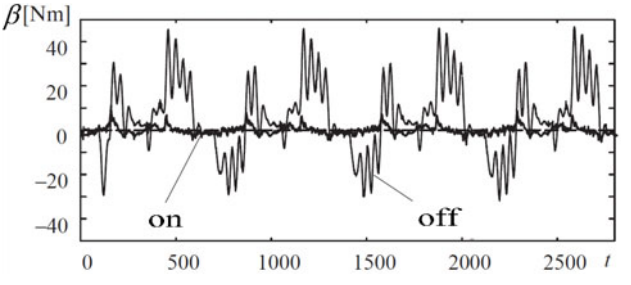


Fig. 9. Pre-stress with null space control on/off

to 3m/s, depending on the acceleration and speed capacity in different parts of the workspace (which is calculated online by the interpolator [16]).

First, it is looked at the effect of feedback null-space control. In a first test, the null-space controller is turned off during the robot movement. Then the same movement is repeated with null space control switched on. Fig. 9 shows the resulting pre-stress, represented by the measured coordinate β: Without null space feedback control, peaks exceeding 40Nm are visible. These can be caused by different friction parameters of the gearboxes and kinematic inaccuracies of the robot structure. Since decoupling of null and motion space is never perfect, oscillations occurring in the motion space can be visible in the null space as well. It is noteworthy that the movement of the robot at high speed is not smooth, emitting an audible humming sound. The structural oscillations seen in the plot are likely to cause excessive wear of the robot bearings.

Whenever the null space controller is turned on (with zero reference value), things change – the pre-stress coordinate remains below 5Nm and the robot movement is free of unexpected sounds. The plot proves the benefits of a feedback-controlled null space in actuation redundant parallel robots.

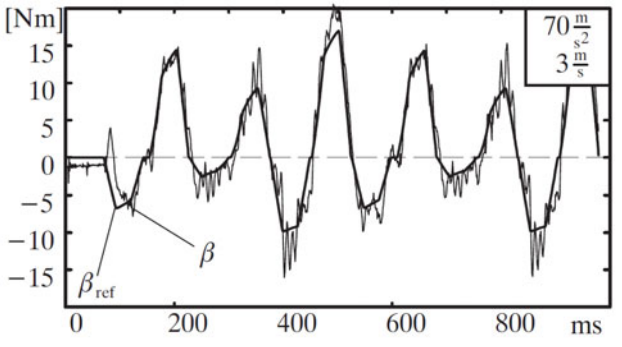


Fig. 10. Tracking a dynamic pre-stress reference

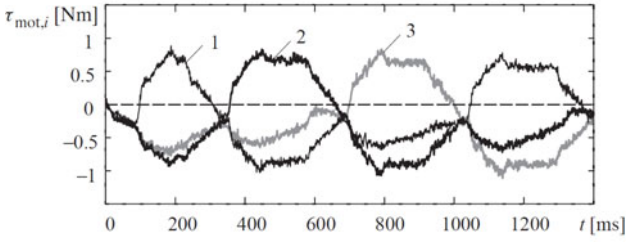


Fig. 11. Motor torques at constant pre-stress of −40Nm

Fig. 10 shows that also dynamic pre-stress references can be tracked quite accurately and with good dynamic response. In the plot, the pre-stress is changed from positive to negative 6 times within one second during movement of the robot, and the stress level is tracked within ±5Nm tolerance.

Fig. 11 shows a plot of the motor torques when a constant pre-stress of −40Nm is applied to the system. During the time when the robot rests in the fine points, a non-zero torque is applied. This creates a pre-stress of gears and bearings during standstill of the robot, which is when usually the highest repeatability is required, especially in pick-and-place and assembly applications. Hence, this control mode establishes an anti-backlash strategy. An improved strategy would be to just ramp up the pre-stress dynamically before a fine point, and apply another redundancy strategy during fast movement.

5.2 Static Tests, Repeatability with Pre-stress

The effect of the anti backlash strategy previously introduced needs to be verified. In a theoretical experiment, we showed that the average repeatability of the manipulator, with the gearbox hysteresis and resolver resolution being the dominant sources of error, is ±51μm and ±3.1μm, respectively [19]. Due to the non-linear robot kinematics, the repeatability is not uniform. Rather, we can define repeatability ellipses

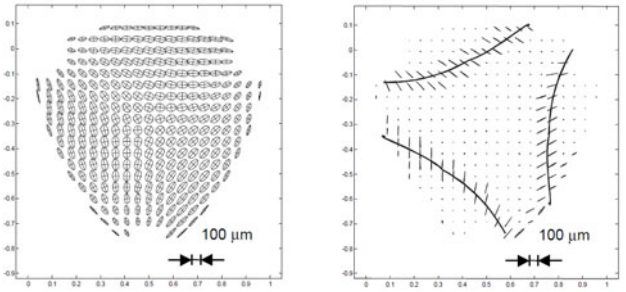


Fig. 12. Repeatability error ellipses without and with pre-stress

defined by the eigenvalues of the Jacobian. Fig. 12 (on the left) shows the repeatability ellipses over the workspace without pre-stress. If the anti-backlash strategy is applied, in the theoretical experiment we assume that the gearbox hysteresis disappears, as the gearbox teeth are always forced into a determined position. Hence the remaining error source in the drive train is the resolution of the motor measurement system. The theoretical remaining error ellipsoids are shown on the right side of Fig. 12.

A repeatability test with a micrometer gauge has been performed on the PA-R-MA prototype to verify these theoretical predictions. We found a repeatability of ±35µm without and less than ±5µm with pre-stress applied, which is close to the predictions. The figure shows also that redundancy control breaks down near the singularity loci of two leg pairs; here, where two of the links B_iC align, pre-stress can only be applied in direction of the links. While the control works seamlessly through the singularity loci, the anti backlash effect gets lost in one direction. This effect could be avoided by adding another redundant chain. Details can be found in [16], [19].

6 Conclusions and Remaining Work

The work with the PA-R-MA manipulator has resulted in a unique test bed for redundancy control investigations. To the authors' best judgment, the following has been achieved for the first time:

- Control of a fast parallel robot with actuation redundancy and gearboxes exceeding 4g acceleration
- In-structure pre-stress measurement with pre-stress feedback control,
- Verification of anti-backlash control strategies and increase of repeatability by a factor of 6
- Decoupled Cartesian and null space control

It was also shown that the null space dynamics for such types of robots can be made largely invariant to the robot pose by choice of a high stiffness ratio between links that experience only compression and traction, and links that experience bending.

This feature facilitates the null space feedback control – in fact, we have achieved a globally stable and high performance control over the complete working range without switching control parameters. Due to time limitations, it was not possible to verify all redundancy strategies known from literature on the prototype. Future work could also address the problem of switching redundancy control objectives along a robot trajectory.

References

1. Buttolo, P., Hannaford, B.: Pen-based force display for precision manipulation in virtual environments. In: Proc. IEEE Virtual Reality Annual Tnt. Symp., North Carolina, pp. 217–224 (1995)
2. Marquet, F., Company, O., Krut, S., Gascuel, O., Pierrot, F.: Control of a 3-dof over-actuated parallel mechanism. In: Proc ASME: Int. Design Eng. Techn. Conf., Montreal (2002)
3. Müller, A.: Stiffness control of redundantly actuated parallel manipulators. In: Proc. IEEE Int. Conf. on Rob. and Autom., Orlando (2006)
4. Yi, B.-J., Freeman, R., Tesar, D.: Force and stiffness transmission in redundantly actuated mechanisms: The case for a spherical shoulder mechanism. Robotics, Spatial Mechanisms and Mechanical Systems 45, 163–172 (1994)
5. Müller, A.: Internal preload control of redundantly actuated parallel manipulators – its application to backlash avoiding control. IEEE Transactions on Robotics 21(4), 668–677 (2005)
6. Bruyninckx, H.: Kinematically dual manipulators. In: Proc. IEEE Int. Conf. on Rob. and Autom., Detroit (1999)
7. Dasgupta, B., Mruthyunjaya, T.: Force redundancy in parallel manipulators: Theoretical and practical issues. Mechanism and Machine Theory 33(6), 727–742 (1998)
8. Lee, J., Yi, B.-J., Oh, S.-R., Suh, I.: Optimal design of a five-bar finger with redundant actuation. In: Proc. IEEE Int. Conf. on Rob. and Autom., Leuven, vol. 3 (1998)
9. Hayward, V., Kurtz, R.: Multiple-goal kinematic optimization of a parallel spherical mechanism with actuator redundancy. IEEE Trans. on Rob. and Autom. 8(5), 644–651 (1992)
10. Nahon, M., Angeles, J.: Force optimization in redundantly-actuated closed kinematic chains. In: Proc. IEEE Int. Conf. on Rob. and Autom., Montreal (1989)
11. So, B., Yi, B.-J., Kim, W., et al.: An in-parallel actuated manipulator with redundant actuators for gross and fine motion. In: Proc. IEEE Int. Conf. on Rob. and Autom., New Orleans (2004)
12. Ropponen, T.: Actuation Redundancy in a Closed-Chain Robot Mechanism. Ph.D. thesis, Acta Polytechnica Scandinavica (1993)
13. Kock, S., Schumacher, W.: A parallel x-y manipulator with actuation redundancy for high-speed and active-stiffness applications. In: Proc. IEEE Int. Conf. on Rob. and Autom., Leuven (1998)
14. Marquet, F., Pierrot, F., Company, O.: A statistical approach for the computation of the forward kinematic model of redundantly actuated mechanisms. In: Proc. IROS 2003, Las Vegas (2003)
15. Yiu, Y., Li, Z.: Optimal forward kinematics map for a parallel manipulator with sensor redundancy. In: IEEE Int. Conf. on Computational Intelligence in Rob. and Autom., Kobe (2003)

16. Kock, S.: Parallelroboter mit Antriebsredundanz. Ph.D. thesis, TU Braunschweig (2001)
17. Kock, S., Schumacher, W.: A mixed elastic and rigid-body dynamic model of an actuation redundant parallel robot with high-reduction gears. In: Proc. IEEE Int. Conf. on Robotics and Autom., San Francisco (2000)
18. Kock, S., Schumacher, W.: Control of a fast parallel robot with a redundant chain and gearboxes: Experimental results. In: Proc. IEEE Int. Conf. on Robotics and Autom., San Francisco (2000)
19. Kock, S.: Genauigkeitssteigerung von Parallelkinematiken durch redundante Antriebe und Rückkopplung der inneren Kräfte. In: VDI Mechatronik-Tagung, Frankenthal (2001)

Robust Gain Scheduling for Smart-Structures in Parallel Robots

Stephan Algermissen and Michael Sinapius

Abstract. Smart-structures offer the potential to increase the productivity of parallel robots by reducing disturbing vibrations caused by high dynamic loads. In parallel robots the vibration behavior of the structure is position dependent. A single robust controller is not able to gain satisfying control performance within the entire workspace. Hence, vibration behavior is linearized at several operating points and robust controllers are designed. Controllers can be smoothly switched by gain-scheduling. A stability proof for fast varying scheduling parameters based on the Small-Gain Theorem is developed. Experimental data from TRIGLIDE, a four degree of freedom (DOF) parallel robot of the Collaborative Research Center 562, validate the presented concepts.

1 Introduction

Since 30 years, smart-structures have been an emerging technology for reduction of structural vibrations in order to increase performance of components and machines. In all kinds of industries, e.g. aerospace, machine and plant manufacture or automobile technology, vibration phenomena are present. These phenomena and their consequences and effects are usually undesired. Therefore, smart-structures technology has a wide a range of application. Nonetheless, each application has to be analyzed precisely. The choice of actuators and sensors as well as their placement are particularly important for the achievable control performance. Superior tasks such as system identification, control synthesis and control can be implemented more generally. Thus, transfer of knowledge from one application to another is possible without much effort. A substantial amount of literature on smart-structures technology is available.

Stephan Algermissen · Michael Sinapius
German Aerospace Center (DLR), Institute of Composite Structures and Adaptive Systems, Lilienthalplatz 7, 38108 Braunschweig, Germany
e-mail: {stephan.algermissen,michael.sinapius}@dlr.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 159–174.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

This paper shows how vibration control is realized for a parallel robot environment according to industrial needs. The main questions of how to synthesize a robust controller and how to gain control performance over the entire workspace using gain-scheduling are answered. Additionally, the methods are applied to parallel robots since their universality enable their use in all kind of vibration control problems.

2 Related Work

Vibration reduction for manipulators is a key application of smart-structures technology. The origins of modeling and control of structures can be found at the end of the seventies and the beginning of the eighties [1]. First, models like simple, flexible arms driven by a rotating motor were regarded. In the majority of cases the studies were limited to the derivation of an analytic model of the dynamic behavior. The studies that followed investigated more complex manipulators with two links [2] and four link mechanisms with flexible arms. KARKOUB und YIGIT modeled a flexible four bar manipulator by analytic equations and simulated the controlled system [3].

The application of smart-structures in parallel robots is little-known in literature. In most cases smart-structures technology is applied to vibration isolation problems with hexapods and Stewart platforms respectively. ANDERSON [4] developed a vibration control for a Stewart platform in the middle of the nineties. It isolated shock sensitive instruments from vibrations of the base. The objective was to isolate and not to position. A current approach for vibration isolation with hexapods is shown by JOSHI in [5]; LQG/LTR and $\mathcal{H}_\infty$ controllers were tested.

The base is the source of disturbance on systems for vibration isolation. For parallel robots, the disturbances result from the inertial forces of the moving robot. Besides the Collaborative Research Center 562, WANG und MILLS from the University of Toronto worked on vibration control of a planar parallel robot with three degrees of freedom (DOF) for several years. The main topic of their work was modeling of the flexible and dynamic behavior of the robot structure [6]. Finite element formulations as well as analytic solutions were proposed. As control strategy they chose modal control [7]. In his PhD thesis WANG [8] describes the strategy in more detail. In spite of a position-dependent formulation of their model, the vibration controller does not function according to the position in workspace. His successor ZHANG is working on the improvement of the coupled flexible and dynamic behavior model for planar parallel robots [9].

3 Robust Control for Smart-Structures

3.1 Introduction

Typical smart-structure systems consist of four main components: the structure, sensors, actuators and the controller. The sensors measure the structure's accelerations, based on these signals the controller calculates the control signals. The signals drive

the actuators to reduce the vibrations. The transfer function from the actuators' input to the output of the sensors is called the controlled plant. In smart-structures technology plants are usually multiple-input-multiple-output (MIMO) systems. Therefore, classical control methods like PID control are not suitable for such applications. Modern robust control algorithms like $\mathcal{H}_2$- and $\mathcal{H}_\infty$-control [10] actually have been able to realize good performance and robustness to plant disturbances. This section shows how $\mathcal{H}_\infty$-control is synthesized for generic smart-structure systems.

Transfer matrices of discrete state-space systems are used with the following notation:

$$\begin{aligned} \boldsymbol{x}(n+1) &= \boldsymbol{A}\boldsymbol{x}(n) + \boldsymbol{B}\boldsymbol{u}(n) \\ \boldsymbol{y}(n) &= \boldsymbol{C}\boldsymbol{x}(n) + \boldsymbol{D}\boldsymbol{u}(n) \end{aligned} \quad \Longleftrightarrow \quad \boldsymbol{G} = \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{C} & \boldsymbol{D} \end{array}\right]$$

with inputs $\boldsymbol{u}$, outputs $\boldsymbol{y}$ and states $\boldsymbol{x}$.

3.2 Generalized Framework for Control Synthesis

Controllers for smart-structures are synthesized for good disturbance reduction. The objective is to minimize the influence of disturbances onto the controlled variable. Fig. 1 shows a generic control loop with plant $\boldsymbol{G}$ and controller $\boldsymbol{R}$. Vector $\boldsymbol{r}$ is the reference input, $\boldsymbol{u}$ the control variable, $\boldsymbol{d}$ the process noise and $\boldsymbol{y}$ the controlled variable. The controlled plant $\boldsymbol{G}$ consists of the actuator inputs, the structure and the sensor outputs. The in- and output relations in Fig. 1 are:

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{SG} \\ \boldsymbol{RS} & -\boldsymbol{RSG} \end{bmatrix} \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{d} \end{bmatrix} \tag{1}$$

with sensitivity $\boldsymbol{S}$ and complementary sensitivity $\boldsymbol{T}$

$$\boldsymbol{S} = [\boldsymbol{E} + \boldsymbol{GR}]^{-1} \qquad \boldsymbol{T} = [\boldsymbol{E} + \boldsymbol{GR}]^{-1}\boldsymbol{GR} = \boldsymbol{E} - \boldsymbol{S} \tag{2}$$

and identity matrix $\boldsymbol{E}$.

In robotic applications vibrations of the structure result from inertial forces induced by acceleration and deceleration of the robot. These disturbing inertial forces are modelled as process noise entering the closed control loop between controller

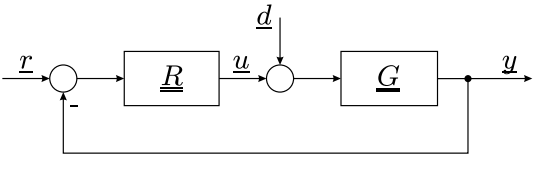


Fig. 1. Closed control loop

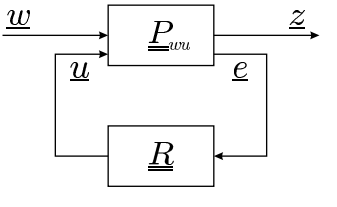


Fig. 2. Generalized framework $\boldsymbol{P}_{wu}$ in LFT form

and plant. In the past this formulation of the problem became the most effective way to realize vibration reduction with good performance [11, 12, 13]. The influences of disturbance $\boldsymbol{d}$ onto controlled variable $\boldsymbol{y}$ are defined by the disturbance transfer function $\boldsymbol{SG}$. To achieve good disturbance rejection the maximum singular value of $\boldsymbol{SG}$ has to be as small as possible. As maximum singular value of a transfer function, the $\mathcal{H}_\infty$-norm of a discrete system is defined as:

$$\|\boldsymbol{G}\|_\infty = \sup_\omega \bar{\sigma}\left(\boldsymbol{G}\left(\mathrm{e}^{j\omega T}\right)\right) \tag{3}$$

with frequency ω, imaginary unit j and sampling time T. Therefore, the objective of the controller is to minimize $\|\boldsymbol{SG}\|_\infty$. As the name implies, $\mathcal{H}_\infty$ control theory is very suitable for this kind of problem. It minimizes the $\mathcal{H}_\infty$ norm of a given performance criterion. The criterion for this application would be the minimization of $\|\boldsymbol{SG}\|_\infty$. But without additional criteria, this optimization would lead to very large control variables ($\boldsymbol{u}$) and saturated actuators. To limit $\boldsymbol{u}$, the $\mathcal{H}_\infty$ norm of the transfer function from $\boldsymbol{r}$ to $\boldsymbol{u}$ $\|\boldsymbol{RS}\|_\infty$ has to be limited. $\boldsymbol{RS}$ affects the transfer function from disturbance $\boldsymbol{d}$ to controlled variable $\boldsymbol{y}$ similarly, see (1). The general formulation of a $\mathcal{H}_\infty$ control problem is done in a generalized framework in the form of a so-called linear fractional transformation (LFT), see Fig. 2.

The objective of the controller synthesis is to find a controller $\boldsymbol{R}$ that minimizes the $\mathcal{H}_\infty$ norm of the transfer function $\boldsymbol{T}_{zw}$ from performance inputs $\boldsymbol{w}$ to the outputs $\boldsymbol{z}$. In practice, the determination of the optimal controller is numerically and theoretically hard. Therefore most of the available algorithms [10] determine a sub-optimal controller that realizes

$$\|\boldsymbol{T}_{zw}\|_\infty < \gamma \tag{4}$$

for a given γ.

For synthesis, the performance in- and outputs have to be defined and afterwards the closed loop in Fig. 1 has to be transformed into LFT form in Fig. 2 by pulling out the controller $\boldsymbol{R}$. Experiments have shown that the plant in (1) leads to good vibration reduction performance ($\boldsymbol{SG}$) and to a smooth controller ($\boldsymbol{RS}$), if it is chosen as $\boldsymbol{T}_{zw}$ [11]. Thus, the performance inputs are $\boldsymbol{d}$ and $\boldsymbol{r}$ and the outputs are $\boldsymbol{y}$ and $\boldsymbol{u}$. With (1) and (4) the objective reads:

$$\left\| \begin{bmatrix} \boldsymbol{T} & \boldsymbol{SG} \\ \boldsymbol{RS} & -\boldsymbol{RSG} \end{bmatrix} \right\|_\infty < \gamma \tag{5}$$

The limitation of the $\mathcal{H}_\infty$ norm of the diagonal terms $\boldsymbol{T}$ and -$\boldsymbol{RSG}$ leads to robustness of the control loop to multiplicative uncertainties of the plant [10].

3.3 Weighting Filters

The further specification of objectives is done with so-called weighting functions W. The inverse of a weighting function describes the desired singular value function of the weighted transfer function. For example, let $\boldsymbol{T}$ be the complementary sensitivity

that shall be shaped. A weighting filter $\boldsymbol{W}_T$ is placed at the system output $\boldsymbol{y}$. With $\gamma = 1$ the control objective is:

$$\|\boldsymbol{T}_{zw}\|_\infty = \|\boldsymbol{W}_T \boldsymbol{T}\|_\infty < 1 \tag{6}$$

The approximation

$$\|\boldsymbol{T}\|_\infty < \|\boldsymbol{W}_T^{-1}\|_\infty \tag{7}$$

shows that the controller must shape the complementary sensitivity close to $\boldsymbol{W}_T^{-1}$.

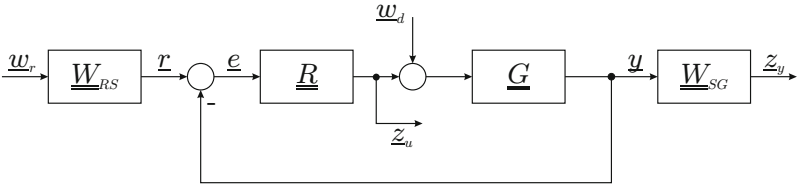


Fig. 3. Closed control loop with weighting functions

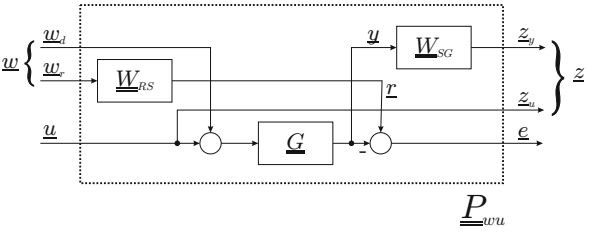


Fig. 4. Generalized framework of weighting scheme

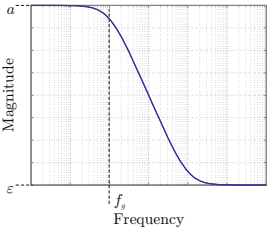


Fig. 5. Frequency response of $W^{-1}(s)$

Applied to (5), two weighting systems $\boldsymbol{W}_{SG}$ and $\boldsymbol{W}_{RS}$ are introduced. $\boldsymbol{W}_{SG}$ is placed at the output $\boldsymbol{y}$ and $\boldsymbol{W}_{RS}$ at the input $\boldsymbol{r}$, see Fig. 3. The weighting systems are diagonal transfer matrices with identical weighting functions on the diagonal. After transformation to LFT form, see Fig. 4, the plant $\boldsymbol{P}_{wu}$ is usable for controller synthesis. The input and output of $\boldsymbol{P}_{wu}$ are:

$$\begin{bmatrix} \boldsymbol{z}_y \\ \boldsymbol{z}_u \\ \boldsymbol{e} \end{bmatrix} = \boldsymbol{P}_{wu} \begin{bmatrix} \boldsymbol{w}_r \\ \boldsymbol{w}_d \\ \boldsymbol{u} \end{bmatrix} \tag{8}$$

with

$$\boldsymbol{P}_{wu} = \left[\begin{array}{c||c|c} \boldsymbol{A} & \boldsymbol{B}_1 & \boldsymbol{B}_2 \\ \hline\hline \boldsymbol{C}_1 & \boldsymbol{D}_{11} & \boldsymbol{D}_{12} \\ \hline \boldsymbol{C}_2 & \boldsymbol{D}_{21} & \boldsymbol{D}_{22} \end{array}\right] = \left[\begin{array}{ccc||c|c} \boldsymbol{A}_G & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}_G & \boldsymbol{B}_G \\ \boldsymbol{0} & \boldsymbol{A}_R & \boldsymbol{0} & \boldsymbol{B}_R & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{B}_S\boldsymbol{C}_G & \boldsymbol{0} & \boldsymbol{A}_S & \boldsymbol{0} & \boldsymbol{B}_S\boldsymbol{D}_G & \boldsymbol{B}_S\boldsymbol{D}_G \\ \hline\hline \boldsymbol{D}_S\boldsymbol{C}_G & \boldsymbol{0} & \boldsymbol{C}_S & \boldsymbol{0} & \boldsymbol{D}_S\boldsymbol{D}_G & \boldsymbol{D}_S\boldsymbol{D}_G \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E} \\ \hline -\boldsymbol{C}_G & \boldsymbol{C}_R & \boldsymbol{0} & \boldsymbol{D}_R & -\boldsymbol{D}_G & -\boldsymbol{D}_G \end{array}\right]$$

The indices G, R and S label the state-space matrices of the plant $\boldsymbol{G}$ and the weighting systems $\boldsymbol{W}_{RS}$ and $\boldsymbol{W}_{SG}$.

Until now the state-space matrices of the weighting filters $\boldsymbol{W}_{SG}$ and $\boldsymbol{W}_{RS}$ have not been determined. Due to their shaping functionality, they are plant-dependent and must therefore be parameterizable. As mentioned above, they are diagonal transfer matrices with identical weighting functions. Their dynamic behavior must not be more complex than needed, because the sum of the states of weighting systems and plant equals the number of controller states. For weighting functions simple, stable and invertible transfer functions are desired. Invertible low-passes are very useful for specifying control objectives in smart-structure applications. A simple continuous approach is:

$$W^{-1}(s) = \frac{\varepsilon s + a 2\pi f_g}{s + 2\pi f_g} \tag{9}$$

The 3 dB cut-off frequency is f_g. The meaning of parameters a and ε becomes apparent by looking at the limits:

$$W^{-1}(s = 0) = a \qquad \lim_{s\to\infty} W^{-1}(s) = \varepsilon$$

Tuning of the filter is possible by tuning three simple parameters. For low-passes $a > \varepsilon$ must hold. By choice of $a < \varepsilon$, a high-pass can be generated. The relation of the parameters to the frequency response of the filter is shown in Fig. 5. Stability of W as well as W^{-1} is given as long as f_g and a/ε are greater than zero. The steepness of the filter response can be increased by series connection of several filters.

Converting to a state-space model and applying the Tustin transformation

$$s = \frac{2}{T}\frac{z-1}{z+1}, \tag{10}$$

which is a bi-linear transformation [14], the discrete state-space representation for the filter is obtained:

$$W(z) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \qquad A = \left(1 - \frac{\pi f_g a T}{\varepsilon}\right) B \qquad B = \frac{\varepsilon}{\varepsilon + \pi f_g a T}$$

$$C = \frac{2\pi f_g T}{\varepsilon}\left(1 - \frac{a}{\varepsilon}\right) B \qquad D = \frac{\pi f_g T}{\varepsilon}\left(1 - \frac{a}{\varepsilon}\right) B + \frac{1}{\varepsilon}$$

The objective can easily be defined with these filter equations. The matrices have to be inserted into $\boldsymbol{P}_{wu}$ in (8), which is then fed into a $\mathcal{H}_\infty$ controller synthesis algorithm.

4 Robust Gain-Scheduling

4.1 Introduction

Vibration behavior of parallel robots is position-dependent. This results in the dependency of $\boldsymbol{G}$ from the position of the effector. Usually, a single controller is not capable of guaranteeing satisfactory performance and stability for the entire workspace. Eigenfrequencies of the plant shift and damping changes when the effector moves from one position to another. This fact led to the development of the robust gain-scheduling control concept for two parallel robots in the Collaborative Research Center 562, FIVE-BAR and TRIGLIDE [15], as proposed in [11, 12]. The chosen robust gain-scheduling approach makes use of several controllers, each valid in a different domain of the robot's workspace. The key is to find a proof of stability for the controller switching, which is necessary since the effector moves freely between the domains. In consequence of the Collaborative Research Center's idea of a stand-alone adaptive system that calibrates itself with automated system identification and control synthesis, the stability proof must be capable of being integrated into the robot's control environment. The environment is a control PC with real-time operating system. Stability analysis therefore has to be implementable in C/C++. Gain-scheduling is one of the most utilized concepts in non-linear control. The basic idea of this concept is 'Divide and conquer!'. Non-linear plants are linearized at a set of equilibrium points. For control synthesis linear control theory is applied to the set of linear plants. The advantage of linear control theory is the existence of an abundance of approved methods for linear systems. Gain-scheduling problems are classified by type of linearization and parameter dependency. A good overview of existing methods is provided in the papers of LEITH [16] and RUGH [17]. The approach used

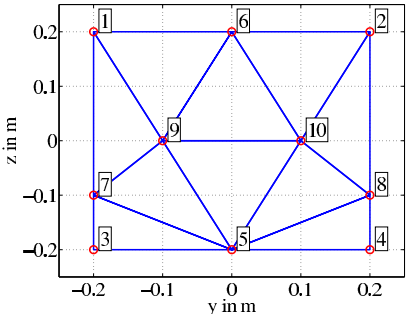


Fig. 6. Workspace decomposition of parallel robot TRIGLIDE

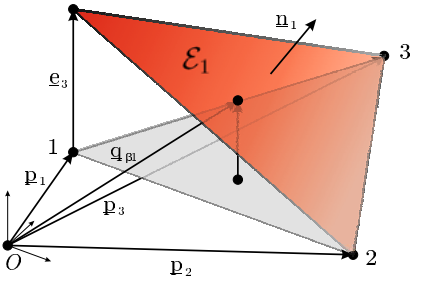


Fig. 7. Factor β_1 as function of position in triangle

in this paper is classifiable with SMALL GAIN LINEAR FRACTIONAL TRANSFORMATION (LFT) GAIN-SCHEDULING and bases on the work of PACKARD [18]. The system is divided into time-variant and -invariant sub-systems. The time-variant parameters are inputs to the plant and the controller and available in realtime. The Small-Gain Theorem [10] describes the robust stability of a system with the $\mathcal{H}_\infty$-norm of the time-variant and -invariant sub-systems. The results of the theorem concerning the system's stability are very conservative [19, 20]. The scaling of the Small-Gain Theorem with a constant matrix is able to reduce this conservatism, as PACKARD [18] and SHAMMA [21] state. The main source of conservatism of Small-Gain Theorem is the assumption that variation speed of time-variant parameters is unlimited. TENG [22] shows that the so-called Swapping Lemma is a possibility to limit this variation speed. Nevertheless, the computational effort is very high and the order of the analyzed system increases dramatically. Another possibility is the assumption of a 'small' parameter variation rate and therefore quasi-static behavior. SHAMMA [23] advises against these simplifying assumptions. For a safe proof of stability the rates should be assumed as infinite.

Workspace decomposition was chosen for a gain-scheduling concept of two of the parallel robots in the Collaborative Research Center 562. Due to several assumptions, vibration behavior of the robots depends on the position of the effector in a planar workspace only [13]. Fig. 6 shows the decomposition of TRIGLIDE's workspace with ten triangles and ten operating points. In each operating point k a system identification extracts a discrete time state-space model $\boldsymbol{G}_k$. For intermediate points an interpolated model $\hat{\boldsymbol{G}}$ is calculated based on the models of the surrounding triangle's corners:

$$\widehat{\boldsymbol{G}} = \sum_{k=1}^{3} \beta_k \, \boldsymbol{G}_k \tag{11a}$$

The factors β_k hold:

$$\sum_{k=1}^{3} \beta_k = 1 \qquad\qquad 0 \;\leq\; \beta_k \leq 1 \tag{11b}$$

Factor β_k is described as a linear function of the effector's position within the triangle. In operating point k β_k equals 1. All remaining β are zero. The first triangle in Fig. 6 is spanned by operating points 1-3. The function of β_1 corresponding to operating point 1 is illustrated in Fig. 7. The function is a plane ($\mathcal{E}_1$) that is spanned by three points. At point 1 with $\beta_1 = 1$ and at points 2 and 3 with $\beta_1 = 0$. This condition guarantees, that the functions of β_k are continuous over all triangle domains in workspace. Calculation of β_1 at intermediate points is done with simple calculus that is computable in realtime.

For each operating point k a robust $\mathcal{H}_\infty$ controller $\boldsymbol{R}_k$ is synthesized. The process of system identification and control synthesis is automatized and executable on TRIGLIDE's control PC [11]. According to the plant, the factors β_k are responsible for determination of the interpolated controller $\widehat{\boldsymbol{R}}$:

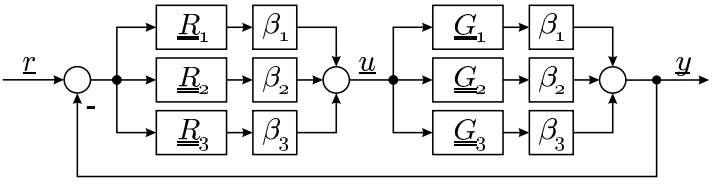


Fig. 8. Block diagram of closed control loop with interpolated controller and plant

$$\widehat{\boldsymbol{R}} = \sum_{k=1}^{3} \beta_k \, \boldsymbol{R}_k \tag{12}$$

The resulting block diagram for a single triangle in workspace with interpolated controller and plant is shown in Fig. 8. All β_k are time-variant and thus stability of the switching process has to be proven for the entire workspace.

4.2 Proof of Stability

The Small-Gain Theorem is used for determination of robust stability in robust control theory. In Fig. 9 the nominal linear plant $\boldsymbol{M}$ is disturbed by perturbations $\boldsymbol{\Delta}$. The external in- and outputs are $\boldsymbol{u}$ and $\boldsymbol{y}$, the internal ones $\boldsymbol{i}$ and $\boldsymbol{o}$. The structure of the perturbations $\boldsymbol{\Delta}$ is not relevant for the Small-Gain Theorem. Furthermore, they may be time-variant or -invariant. The systems of the general loop are:

$$\begin{bmatrix} \boldsymbol{o} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{i} \\ \boldsymbol{u} \end{bmatrix} \qquad \boldsymbol{i} = \boldsymbol{\Delta o} \tag{13}$$

The loop in Fig. 9 represents an upper linear fractional transformation (LFT). After elimination of the inner in- and outputs $\boldsymbol{i}$ und $\boldsymbol{o}$ the transfer function of the entire system is:

$$\boldsymbol{y} = \left[\boldsymbol{M}_{21} \left[\boldsymbol{E} - \boldsymbol{\Delta M}_{11} \right]^{-1} \boldsymbol{\Delta M}_{12} + \boldsymbol{M}_{22} \right] \boldsymbol{u} \tag{14}$$

The system is stable, if

$$\det \left(\boldsymbol{E} - \boldsymbol{\Delta M}_{11} \right) \neq 0 \tag{15}$$

holds. This condition is formulated by Small-Gain Theorem based on $\mathcal{H}_\infty$-norms of the sub-systems:

Theorem 1. *Small Gain Theorem*
Systems $\boldsymbol{M}$ and $\boldsymbol{\Delta}$ are stable and $\gamma > 0$. Loop in Fig. 9 is stable, if

(a) $\|\boldsymbol{\Delta}\|_\infty \leq 1/\gamma$

(b) $\|\boldsymbol{M}_{11}\|_\infty < \gamma$

holds.

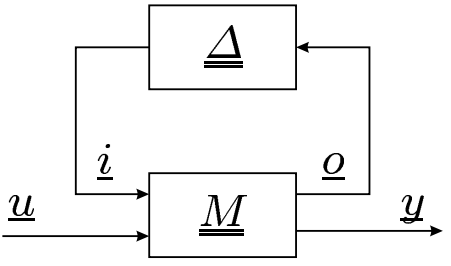


Fig. 9. LFT for robust stability analysis

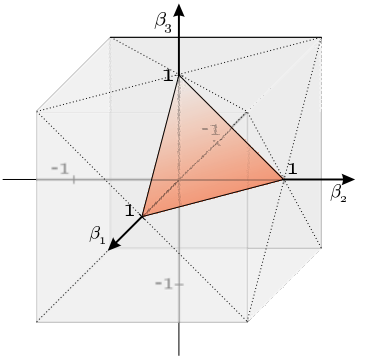


Fig. 10. Real and allowed intervals of $\beta_1 - \beta_3$

Results of Small-Gain Theorem are very conservative, but are universal and easy to use. No structure of $\boldsymbol{\Delta}$ has to be defined yet. Perturbations could even have time-variant non-linear behavior.

For application of the Small-Gain Theorem to the stability problem, the block diagram in Fig. 8 has to be transformed into the general loop form of Fig. 9. The time-variant factors β_k are separated and placed in the time-variant perturbation matrix $\boldsymbol{\beta}$:

$$\boldsymbol{\Delta} = \boldsymbol{\beta} = \text{diag}\,(\beta_1\boldsymbol{E},\, \beta_1\boldsymbol{E},\, \beta_2\boldsymbol{E},\, \beta_2\boldsymbol{E},\, \beta_3\boldsymbol{E},\, \beta_3\boldsymbol{E})_{3(n_u+n_y)\times 3(n_u+n_y)} \tag{16}$$

Dimensions of in- and output vectors $\boldsymbol{u}$ and $\boldsymbol{y}$ are n_u and n_y. The remaining time-invariant part of the system is combined in matrix $\boldsymbol{M}$. Due to (11b) the $\mathcal{H}_\infty$-norm of $\boldsymbol{\beta}$ is limited to 1. This leads to $\gamma = 1$ in Small-Gain Theorem.

To prove stability for switching processes in robot's workspace, the theorem above must hold for all triangles. As expected, experiments showed that the theorem gives very conservative results. Vibration reduction is poor as a result of low authority controllers. Four main sources of conservatism of Small-Gain Theorem can be identified:

1. Structure of $\boldsymbol{\beta}$
   The Small-Gain Theorem limits the $\mathcal{H}_\infty$-norm of $\boldsymbol{\beta}$. The diagonal structure is left disregarded. $\boldsymbol{\beta}$ is treated as unstructured perturbation.
2. Real β_k
   The Small-Gain Theorem assumes that perturbations are complex valued in general. But $\boldsymbol{\beta}$ is real. A β_1 of j for example corresponds to a positive feedback loop.
3. Linear dependence of β_k
   The factors β_k are not linear independent. They are coupled by (11b) such that their sum equals 1. The norm's upper limit of 1 allows linear independent β_k in the interval [-1; 1]. To visualize the difference, the range of values for $\beta_1 - \beta_3$ are plotted in Fig. 10. The triangular plane includes all real values that are actually used for β_k due to the constraints in (11b). The cube is the range of values in

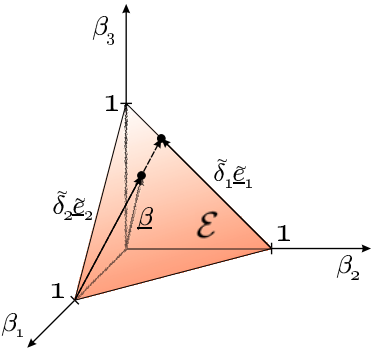


Fig. 11. Coordinate system of plane $\mathcal{E}$

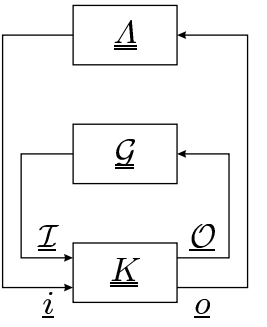


Fig. 12. LFT of closed loop for stability analysis

which the β_k may float freely as consequence of the $\mathcal{H}_\infty$-norm formulation. The fact that β_k even may be complex is left unattended here.

4. Parameter variation rate
   In the Small-Gain Theorem no assumptions of parameter variation rates are made. Theoretically the rates may be infinite. The system is stable if the theorem provides it. In reality the variation rates of the robot's system vary with limited rates, so that the theorem gives quite conservative results.

By transformations on $\boldsymbol{\beta}$, derived below, the conservatism can be reduced significantly.

According to item 3 of the list, the linear dependence of all β_k is left unattended. The triangular plane spanned in Fig. 10 is modelled inadequately by three β_k and equation (11b). Only two coordinates are needed to model a plane correctly. In this special case where the plane is triangular, a common plane equation like the normal form is not reasonable. Here, the plane is described with a set of non-linear equations involving two independent parameters. A triangular coordinate system is spanned in the triangle by two new base vectors $\tilde{\boldsymbol{e}}_1$ and $\tilde{\boldsymbol{e}}_2$[1], where $\tilde{\boldsymbol{e}}_2$ is a function of $\tilde{\boldsymbol{e}}_1$. The two independent parameters introduced are $\tilde{\delta}_1$ and $\tilde{\delta}_2$. The equations are:

$$\boldsymbol{\beta} = \boldsymbol{e}_1 + \tilde{\delta}_2\tilde{\boldsymbol{e}}_2 \qquad 0 \le \tilde{\delta}_1, \tilde{\delta}_2 \le 1$$
$$\tilde{\boldsymbol{e}}_1 = -\boldsymbol{e}_2 + \boldsymbol{e}_3 \qquad \tilde{\boldsymbol{e}}_2 = -\boldsymbol{e}_1 + \boldsymbol{e}_2 + \tilde{\delta}_1\tilde{\boldsymbol{e}}_1$$

Vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are unit vectors of the first and second dimension. A combination of the equations leads to:

$$\boldsymbol{\beta} = \begin{bmatrix} 1 - \tilde{\delta}_2 \\ \tilde{\delta}_2(1 - \tilde{\delta}_1) \\ \tilde{\delta}_2\tilde{\delta}_1 \end{bmatrix} \tag{17}$$

With this transformation the conservatism caused by reasons described in item 3 is reduced drastically.

[1] The vectors $\tilde{\boldsymbol{e}}_k$ must not have unit length!

The interval of β_k and $\tilde{\delta}_k$ is [0; 1]. A useful interval is [-1; 1], see item 2. Therefore the following linear transformation of $\tilde{\delta}_k$ is essential:

$$\tilde{\delta}_1 = \frac{\delta_1 + 1}{2} \qquad \tilde{\delta}_2 = \frac{\delta_2 + 1}{2} \qquad -1 \leq \delta_1, \delta_2 \leq 1$$

New variables δ_k are introduced here. Applied to (17), the final transformation reads:

$$\boldsymbol{\beta} = \begin{bmatrix} \frac{1}{2}(1 - \delta_2) \\ \frac{1}{4}(\delta_2 + 1)(1 - \delta_1) \\ \frac{1}{4}(\delta_2 + 1)(1 + \delta_1) \end{bmatrix} \tag{18}$$

With this transformation the triangular plane in Fig. 10 was successfully described with two independent parameters δ_1 and δ_2 with interval [-1; 1]. In the following, this non-linear transformation has to be integrated into the linear closed control loop. After transformation perturbation matrix $\boldsymbol{\Delta}$ reduces to:

$$\boldsymbol{\Delta} = \operatorname{diag}(\delta_1 \boldsymbol{E},\ \delta_1 \boldsymbol{E},\ \delta_2 \boldsymbol{E},\ \delta_2 \boldsymbol{E})_{4n_u \times 4n_u} \tag{19}$$

The interpolated controller $\hat{\boldsymbol{R}}$ is calculated with equations (12) and (18) as follows:

$$\hat{\boldsymbol{R}} = \frac{1}{4}\left[4\boldsymbol{R}_1 + (\delta_2 + 1)\left[-2\boldsymbol{R}_1 + \boldsymbol{R}_2 + \boldsymbol{R}_3 + \delta_1(\boldsymbol{R}_3 - \boldsymbol{R}_2)\right]\right] \tag{20}$$

The derivation of the interpolated plant $\hat{\boldsymbol{G}}$ is analog to the controller one:

$$\hat{\boldsymbol{G}} = \frac{1}{4}\left[2(1 - \delta_2)\boldsymbol{G}_1 + [1 + \delta_2 - \delta_1(\delta_2 + 1)]\ \boldsymbol{G}_2 + [1 + \delta_2 + \delta_1(\delta_2 + 1)]\ \boldsymbol{G}_3\right] \tag{21}$$

With equations (20) and (21) the block diagram of the closed loop is built with interpolated controller and plant in Fig. 13. Although the interpolated systems have non-linear terms (e.g. $\delta_1\delta_2$) it is possible to separate the δ_k from the closed loop and derive a linear $\boldsymbol{M}$. Additionally all state-space models are collected in a large diagonal model for fast computation of $\boldsymbol{M}$ on the target control PC:

$$\mathcal{G} = \operatorname{diag}(\boldsymbol{R}_1,\ \boldsymbol{R}_2,\ \boldsymbol{R}_3,\ \boldsymbol{G}_1,\ \boldsymbol{G}_2,\ \boldsymbol{G}_3)_{3(n_u+n_y)\times 3(n_u+n_y)} \tag{22}$$

As final result of the transformations, the closed loop block diagram for stability analysis as LFT is shown in Fig. 12. The static connections are collected in the connection matrix $\boldsymbol{K}$. It can be treated as a state-space model with matrix $\boldsymbol{D}$ but

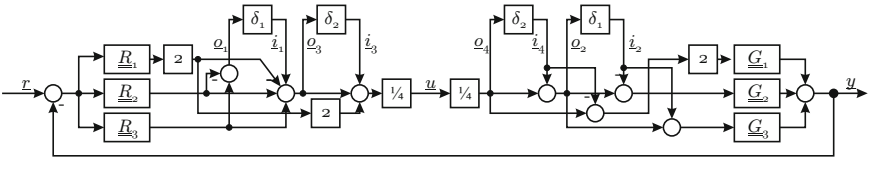


Fig. 13. Block diagram of with interpolated controller and plant

empty matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$. Matrix $\boldsymbol{M}$ needed for analysis is easily calculated with an upper LFT:

$$\boldsymbol{M} = \mathcal{F}_u(\mathcal{G}, \boldsymbol{K}) \tag{23}$$

The source of conservatism discussed in item 2 could be partly reduced by the transformation presented. Only the fact that δ_k could be complex is still disregarded.

The next step concerns item 1. The structure of $\boldsymbol{\Delta}$ has not been taken into account yet. For consideration of its diagonal structure, a method from μ-analysis [24, 25] is adapted. A scaling matrix $\boldsymbol{D}$ is introduced whose structure corresponds to the structure of the perturbation matrix $\boldsymbol{\Delta}$. Usually a scaling matrix can be found such that:

$$||\boldsymbol{D}\boldsymbol{M}\boldsymbol{D}^{-1}||_\infty < ||\boldsymbol{M}||_\infty \tag{24}$$

It is a well known fact that this so-called scaled Small-Gain Theorem gives less conservative results. The base of this theorem is the identity:

$$\boldsymbol{D}^{-1}\boldsymbol{D}\boldsymbol{\Delta} = \boldsymbol{D}^{-1}\boldsymbol{\Delta}\boldsymbol{D} \tag{25}$$

This identity is only valid if $\boldsymbol{D}$ and $\boldsymbol{\Delta}$ commute. They only commute if $\boldsymbol{D}$ has a corresponding structure. In the case of equation (19) the scaling matrix must have the following structure:

$$\boldsymbol{D} = \mathrm{diag}(\boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{D}_3, \boldsymbol{D}_4) \qquad \boldsymbol{D}_k \in \mathbb{C}^{n_u \times n_u} \tag{26}$$

For time-variant perturbations such as $\boldsymbol{\Delta}$ the scaling matrix $\boldsymbol{D}$ must be static and not frequency dependent because time-variant and -invariant systems do not commute [22]. If $\boldsymbol{D}$ is a static, complex matrix the scaled system would be a state-space model with complex entries in its matrices $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$. Algorithms on the control PC are not capable to treat such systems. Therefore it is essential to get a real scaling matrix. For this a step back must be done. For perturbation $\boldsymbol{\Delta}$ a definition as block diagonal, complex matrix still holds:

$$\boldsymbol{\Delta} = \mathrm{diag}(\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2, \boldsymbol{\Delta}_3, \boldsymbol{\Delta}_4) \qquad \boldsymbol{\Delta}_k \in \mathbb{C}^{n_u \times n_u} \tag{27}$$

The corresponding scaling matrix must have the following structure to be commutative [25, 26]:

$$\boldsymbol{D} = \mathrm{diag}(d_1\boldsymbol{E}_{n_u \times n_u}, d_2\boldsymbol{E}_{n_u \times n_u}, d_3\boldsymbol{E}_{n_u \times n_u}, d_4\boldsymbol{E}_{n_u \times n_u}) \qquad d_k \in \mathbb{R},\ d_k > 0 \tag{28}$$

Using this real scaling, the conservatism in results of the stability proof can be reduced again. The optimal scaling matrix $\boldsymbol{D}$ is calculated with routines from μ-analysis. The matrix $\boldsymbol{M}$ - which is an input for μ-analysis - is frequency-dependent and $\boldsymbol{D}$ not. A suitable frequency must be found to calculate a scaling matrix that decreases the $\mathcal{H}_\infty$-norm for the entire spectrum. Tests showed that the frequency where the singular value of $\boldsymbol{M}$ reaches its maximum is very suitable for determination of $\boldsymbol{D}$.

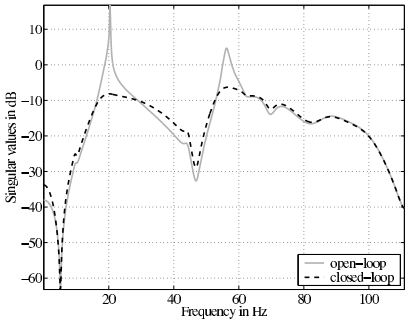


Fig. 14. FIVE-BAR: Maximum singular value of open loop (***G***) and closed loop (***SG***)

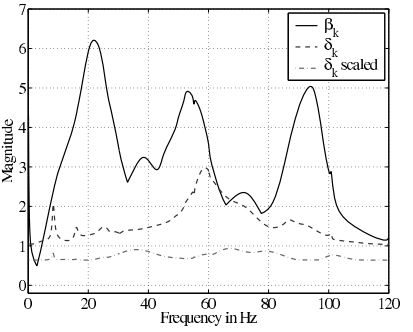


Fig. 15. TRIGLIDE: Maximum singular values of ***M*** for different approaches

The mapping of the triangle controllers and plants to $\boldsymbol{R}_{1-3}$ and to $\boldsymbol{G}_{1-3}$ is not defined. All six possibilities of mapping are equivalent and must be tested. The mapping with the best result is collected for stability test.

4.3 Experiments and Results

Fig. 14 provides an example for the vibration reduction achieved. Open and closed loop maximum singular values from ***G*** and ***SG*** of parallel robot FIVE-BAR [27] are compared. The first and the second eigenfrequencies are significantly reduced by 25 dB and 11 dB. In time domain this result equals an decrease of the settling time of the effector at the end of a trajectory by factor 10.

To illustrate reduced conservatism by the new stability proof, Fig. 15 shows three singular value plots of ***M*** for an arbitrary triangle in TRIGLIDE's workspace. ***M*** is built in three different ways: first using parameters β_k according to Fig. 8, second with new parameters δ_k based on Fig. 13, and third with scaling matrix included. It is obvious how the first method with β_k underestimates the stability of the switching process with a $\mathcal{H}_\infty$-norm of 6.21. The transformation to δ_k enhances the results with a norm of 2.97. The introduction of a suitable scaling matrix finally gives the desired result of 0.93. The stability is therefore proven for this triangle.

5 Conclusion

This article has presented a generalized framework for synthesizing robust controllers for smart-structure applications in parallel robots. The weighting scheme in the form of a discrete state-space system is formulated using easy-to-tune, low-pass shaping filters. The experimental results show a significant reduction of disturbing vibrations.

A new controller switching algorithm with appropriate stability proof applied to vibration control of parallel robots has been developed. The robot's workspace

is divided into triangular domains. For each domain a stability test must be accomplished. This stability test is based on the Small-Gain Theorem. With several transformations on the time-variant switching factors β_k and the introduction of a scaling matrix, the Small-Gain Theorem provides satisfying non-conservative results. Through further investigation, the stability proof can be extended to higher dimensional problems than a planar workspace.

Acknowledgements. This work was funded by the German Research Foundation (DFG) within the scope of the Collaborative Research Center 562 - 'Robotic Systems for Handling and Assembly'.

References

1. Book, W.J.: Controlled motion in an elastic world. Journal of Dynamic Systems, Measurement, and Control 115(2B), 252–261 (1993)
2. Adams, R.J., Apkarian, P., Chrétien, J.-P.: Robust control approaches for a two-link flexible manipulator. In: 3rd International Conference on Dynamics and Control of Structures in Space, pp. 101–116 (1996)
3. Karkoub, M., Yigit, A.S.: Vibration control of a four-bar mechanism with a flexible coupler link. Journal of Sound and Vibration 222(2), 171–189 (1999)
4. Anderson, E.H., Leo, D.J., Holcomb, M.D.: Ultraquiet platform for active vibration isolation. In: Chopra, I. (ed.) Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 2717, pp. 436–451 (1996)
5. Joshi, A., Kim, W.-J.: Modeling and multivariable control design methodologies for hexapod-based satellite vibration isolation. Journal of Dynamic Systems, Measurement, and Control 127(4), 700–704 (2005)
6. Wang, X., Mills, J.: A fem model for active vibration control of flexible linkages. In: Proc. of IEEE International Conference on Robotics and Automation (ICRA), vol. 5, pp. 1050–4729 (2004) ISSN 1050-4729
7. Wang, X., Mills, J.K.: Active control of configuration-dependent linkage vibration with application to a planar parallel platform. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 4327–4332 (2005)
8. Wang, X.: Dynamic Modeling, Experimental Identification, and Active Vibration Control Design of a Smart Parallel Manipulator. Ph.D. thesis, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto (2005)
9. Zhang, X., Mills, J., Cleghorn, W.: Dynamic modeling and experimental validation of a 3-prr parallel manipulator with flexible intermediate links. Journal of Intelligent and Robotic Systems 50(4), 323–340 (2007)
10. Zhou, K., Doyle, J., Glover, K.: Robust and Optimal Control. Prentice-Hall, Englewood Cliffs (1996)
11. Algermissen, S., Rose, M., Keimer, R., Monner, H.P., Breitbach, E.: Automated synthesis of robust controllers for smart-structure applications in parallel robots. In: Proc. of AIAA/ASME/AHS Adaptive Structures Conference, Honolulu, USA (2007)
12. Algermissen, S., Rose, M., Keimer, R., Monner, H.P., Sinapius, M.: Vibration control for smart parallel robots using robust gain-scheduling. In: Borgmann, H. (ed.) ACTUATOR - International Conference on New Actuators, pp. 429–432. Hanseatische Veranstaltungs GmbH, Bremen (2008)

13. Algermissen, S., Rose, M., Keimer, R., Sinapius, M.: Robust gain-scheduling for smart-structures in parallel robots. In: 16th Annual International Symposium on Smart Structures and Materials, San Diego, CA, USA (2009)
14. Safonov, M.G.: Imaginary-axis zeros in multivariable hinf optimal control. In: Curtain, R.F. (ed.) Proc. NATO Advanced Research Workshop on Modeling, Robustness and Sensitivity Reduction in Control Systems, Modelling, Robustness and Sensitivity Reduction in Control Systems, pp. 71–81. Springer, Berlin (1987)
15. Schütz, D., Budde, C., Raatz, A., Hesselbach, J.: Parallel Kinematic Structures of SFB 562. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 109–124. Springer, Heidelberg (2010)
16. Leith, D.J., Leithead, W.E.: Survey of gain-scheduling analysis and design. International Journal of Control 73(11), 1001–1025 (2000)
17. Rugh, W.J., Shamma, J.S.: Research on gain scheduling. Automatica 36(10), 1401–1425 (2000)
18. Packard, A.: Gain scheduling via linear fractional transformations. Systems & Control Letters 22(2), 79–92 (1994)
19. Braatz, R.D., Morari, M.: On the stability of systems with mixed time-varying parameters. Journal of Robust and Nonlinear Control 7, 105–112 (1997)
20. Paganini, F.: Robust stability under mixed time-varying, time-invariant and parametric uncertainty. *Automatica* 32(10), 1381–1392 (1996)
21. Shamma, J.S.: Robust stability with time-varying structured uncertainty. IEEE Transactions on Automatic Control 39(4), 714–724 (1994)
22. Teng, J.: Robust Stability and Performance Analysis with Time-Varying Perturbations. Master's thesis, UC Berkeley, Berkeley, USA (1991)
23. Shamma, J.S., Athans, M.: Gain scheduling: potential hazards and possible remedies. IEEE Control Systems Magazine 12(3), 101–107 (1992) ISSN 0272-1708
24. Lu, W.-M., Zhou, K., Doyle, J.: Stabilization of linear systems with structured perturbations. Technical Report CIT-CDS 93-014, California Institute of Technology, Pasadena, CA, USA (1993)
25. Packard, A., Doyle, J.: The complex structured singular value. Automatica 29(1), 71–109 (1993)
26. Dullerud, G.E., Paganini, F.G.: A course in robust control theory: a convex approach. Springer, New York (1999)
27. Keimer, R., Sinapius, M.: Design and Implementation of Adaptronic Robot Components. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 413–427. Springer, Heidelberg (2010)

Configuration Switching for Workspace Enlargement

Christoph Budde, Manfred Helm, Philipp Last,
Annika Raatz, and Jürgen Hesselbach

Abstract. In order to facilitate a wider use of parallel robots and to make use of their inherent advantages, the drawbacks to these structures, such as singularities in the workspace or the small ratio of workspace to installation space, need to be diminished. The latter can be tackled using an approach presented in this article. An overall workspace, which is substantially larger than the original one, can be obtained by using several robot-configurations corresponding to different working and assembly modes (different solutions of the inverse and direct kinematic problem). The general procedure to change between the configurations requires the passing through singularities and is described exemplarily first for a simple planar manipulator and subsequently for a more complex spatial parallel structure.

1 Introduction

Kinematic structures with closed kinematic chains are known to feature some promising attributes. Since many of these so called parallel structures allow for the

Christoph Budde
ABB Corporate Research Center Germany, Wallstadter Straße 59, 68526 Ladenburg, Germany
e-mail: christoph.budde@de.abb.com

Manfred Helm
Volkswagen AG, Dept. 1564, Berliner Ring 2, 38442 Wolfsburg, Germany
e-mail: manfred.helm@volkswagen.de

Philipp Last
Siemens AG, Dept. E D MV C R&D 4, Nonnendammallee 104, 13629 Berlin, Germany
e-mail: philipp.last@siemens.com

Annika Raatz · Jürgen Hesselbach
Technische Universität Braunschweig, Institute of Production Automation and Machine Tools,
Langer Kamp 19b, 38106 Braunschweig, Germany
e-mail: {a.raatz,j.hesselbach}@tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 175–189.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

drives to be fixed to the base, the moved masses can be kept considerably lower compared to serial structures. This results in high possible accelerations, leading to short cycle times, which are especially interesting for high speed handling tasks. Nevertheless only a few parallel structures are in commercial use so far. This is due to some drawbacks going along with closed kinematic chains. Compared to serial structures the probably most important handicap is a poor ratio of workspace to installation space.

One way to solve this problem consists in the use of several workspaces going along with different solutions of the direct kinematic problem (DKP) and of the inverse kinematic problem (IKP). Usually only one of these workspaces is used [1]. The solutions of the DKP allow to calculate the pose (position and orientation) of the end-effector for given drive positions. They are also known as assembly modes [2]. These assembly modes are usually devided in singularities of type 1 and type 2. (Some exceptions to this rule exist [3], which have no influence, though, on the work presented in this article.) In singularities of type 2 an infinitesimal movement of the end-effector is possible, while all actuators are in rest [4]. On the other hand the solutions of the IKP provide the necessary drive positions for a desired pose of the robot's end-effector. They are known as working modes [5]. I.e. the joint space is divided by singularities of type 1, which occur when a kinematic chain of the structure is either in a stretched position or the links of the chain are folded upon each other [4]. In such a singular position at least one actuator can fulfill an infinitesimal motion without moving the end-effector. Each combination of these working and assembly modes (called configuration here) has a corresponding workspace, called configuration workspace.

Combining several configuration workspaces to a composite workspace is the challenge described in this article. It is shown by means of two examples, the PARAPLACER-robot and the TRIGLIDE-robot, how this challenge can be met. In parallel, another group of researchers is working on this topic, following a similar approach [6].

2 PARAPLACER-Robot

The PARAPLACER-robot [7, 8, 9] depicted in Fig. 1 is a hybrid manipulator with a two degree of freedom (DOF) planar parallel base structure and a serially attached ball-screw-spline allowing for an endeffector movement in z-axis-direction and free rotation around the same axis. The parallel base kinematics PRRRP allows to position the endeffector in a plane. It is actuated by two linear belt-drives. The robot reaches maximum velocities of 3 m/s and maximum accelerations of 2 g with a payload of 3 kg [7]. The following remarks consider the planar base structure only.

2.1 Possible Configurations

The number of possible configurations for the PARAPLACER-structure results from the following considerations: In general there are two different IKP-solutions in

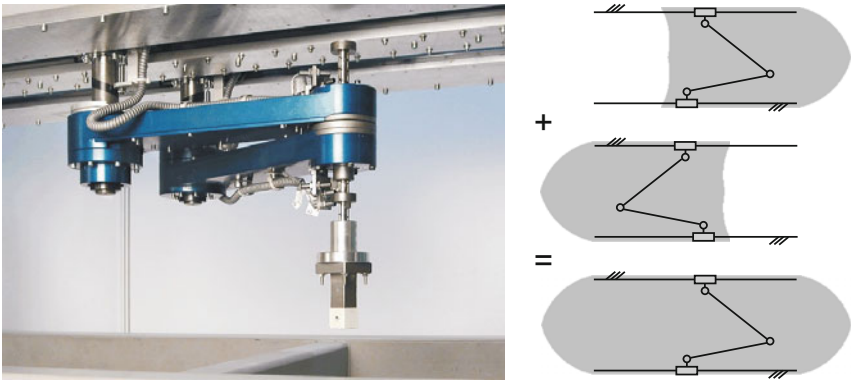


Fig. 1. Prototype of the PARAPLACER-robot and corresponding workspaces.

each of the kinematic chains $i = 1, 2$, meaning that for a given endeffector-position two possible actuator positions can be found. They are distuingished by a binary IKP-configuration-parameter $k_{IKP,i} = \{-1, +1\}$. Both IKP-configuration-parameters are combined in a vector $\mathbf{k}_{IKP} = [k_{IKP,1}, k_{IKP,2}]$. Furthermore, for a given set of drive positions there are two possibilities for the position of the end-effector (2 assembly modes) denoted by $k_{DKP} = \{-1, +1\}$, respectively. These two possibilities result from the solution of the DKP, which can be derived analytically by intersecting two circles [7]. Thus a complete configuration can be uniquely described using the vector $\mathbf{k} = [k_{IKP,1}, k_{IKP,2}, k_{DKP}] = [\mathbf{k}_{IKP}, k_{DKP}]$. $2^3 = 8$ different configurations are resulting from the combination of all possible working and assembly modes. However, only 6 of them are physically existent.

2.2 Workspace Enlargement

Each combination of working and assembly modes is associated with a corresponding workspace. Fig. 1 shows the two largest of these configuration workspaces for the PARAPLACER-structure. They are actually used for working and thus their corresponding configurations are called working-configurations. The two working-configurations are denoted as $[+1, +1, -1]$ and $[-1, -1, +1]$. If it is possible to change between these two working-configurations a resulting composite workspace can be achieved (also shown in Fig. 1), which is significantly bigger. However, several other configuration workspaces and the separating singularities have to be passed to accomplish this change. These configurations are called transition configurations.

One possible path leading from one to the other working configuration through some transition configurations is shown in Fig. 2. It starts from configuration $[+1, +1, -1]$ and consists of three principle steps:

1. **Working mode change**
   In a first step a switch to configuration $[-1, +1, -1]$ is performed. A singularity of type 1 needs to be passed, because the working mode of the first kinematic chain

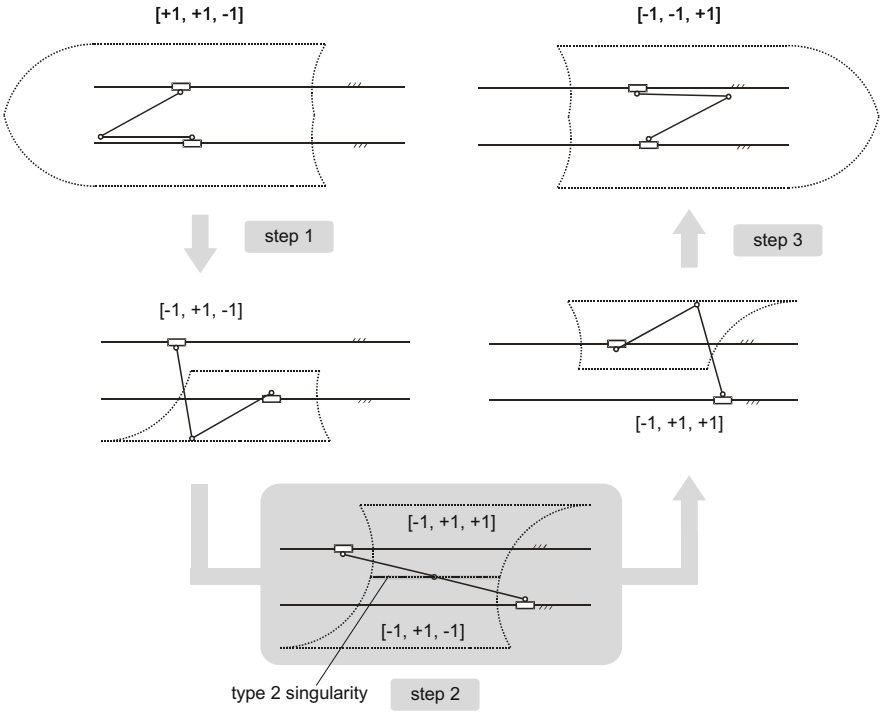


Fig. 2. Possible configuration change sequence for the PARAPLACER-robot.

changes. Such a passing can be managed easily using joint space interpolation [1]; it is no problem from the control point of view.

2. **Assembly mode change**
   A singularity of type 2 needs to be passed in the second step of the changing sequence in order to switch from configuration $[-1, +1, -1]$ to $[-1, +1 + 1]$. In the case of the PARAPLACER-robot such a singularity occurs if the two passive links of the structure are colinear as depicted in the gray-shaded picture corresponding to step 2 in Fig. 2. Passing a singularity of type 2 separates different assembly modes of the structure. This is much more complicated and needs special considerations, since the platform's position cannot be controlled by the drives in such a singularity. Usually it is strictly avoided to guide a parallel mechanism to a singularity of type 2 or even to approach this special kinematic configuration in order to avoid any risk of losing control and damaging the manipulator [10]. A special approach [1, 11] solves this problem. The fundamental idea is to temporarily underactuate the system by deactivating one of the robot's actuators while controlling the other actuators and using some driving force to guide the structure through a singularity. Different driving forces can be used. Inertia forces are used in the implementation applied to the PARAPLACER. That means, one of the actuators is moved so that the robot is accelerated towards the singularity thereby storing kinetic energy. When the other motor is switched idle the inertia of mass

is sufficient to let the mechanism move into the new configuration. Due to the underactuation and the fact that no kinematic models have to be solved throughout this singularity-passing-process, the robot's control is kept free of any critical states and the assembly mode can be changend.

3. **Working mode change**
   Once the singularity of type 2 has been passed and configuration $[-1, +1, +1]$ has been reached, the third and last step in the described configuration-changing sequence consists in another working mode change. As stated above this is not a problem from a control point of view if the robot is controlled in joint space. Thus, the working configuration $[-1, -1, +1]$ can be reached so that both working configurations are indeed combined and a composite workspace can be used which is approximately twice as large as one of the two symmetrical workspaces associated to the working configurations.

3 TRIGLIDE-Robot

The TRIGLIDE-robot [12, 9], see Fig. 3, is another robot, which is capable of changing the configuration in order to enlarge the usable workspace. It is based on the well known Linear-Delta-architecture. Three equally designed kinematic chains connect the end-effector-platform with the base. Each chain is actuated by a direct linear drive. Due to the use of two parallel rods in the build-up of the three chains, the end-effector-platform is always kept at constant orientation. This fully parallel structure features three DOF allowing for free positioning of the end-effector in space. An additional rotation around the z-axis can be accomplished by attaching a serial rotational axis to the platform. Thus the robot is able to perform Schönflies-motions, which are very convenient for pick and place and common assembly operations. Since the rotational axis is irrelevant to the topic discussed here, it is disregarded in the following considerations.

3.1 Possible Configurations for TRIGLIDE-*Structure*

Again, the single configurations of the TRIGLIDE-robot can be uniquely described by a vector $\mathbf{k} = [\mathbf{k}_{IKP}, k_{DKP}]$ with k_{DKP} being a parameter, which indicates the current assembly mode, and $\mathbf{k}_{IKP} = [k_{IKP,1}, k_{IKP,1}, k_{IKP,3}]$ being a vector, which defines the working mode of each of the three kinematic chains. In general two different DKP-solutions exist for a given set of actuator coordinates so that the assembly mode can again be expressed by means of a binary parameter $k_{DKP} = \{-1, +1\}$. Also each of the $i = 1, 2, 3$ working mode parameters $k_{IKP,i}$ is binary, because each drive can generally be positioned in two different ways for a given end-effector position. The combination of all possible working and assembly modes then yields $2^4 = 16$ different configurations. Noting that two of these 16 configurations are not physically existent, the robot can be operated in 14 different configurations.

The shape of the configuration-workspaces and the structure's kinematic characteristics are independent of the end-effector's position along the x-axis (parallel

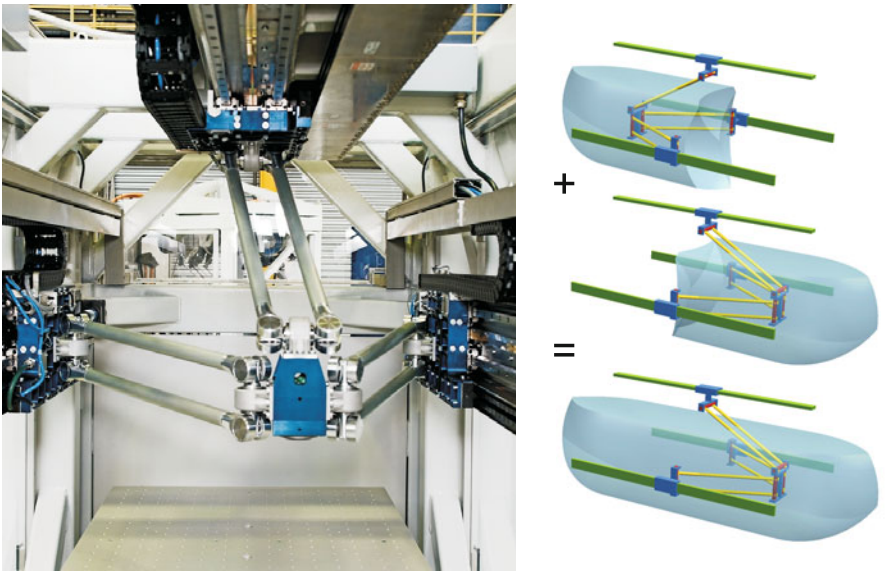


Fig. 3. Prototype of the TRIGLIDE-robot and corresponding workspaces.

to the linear guides, see Fig. 3) due to the parallel arrangement of the linear motors. The workspaces' shapes are changed at the end of their x-extension, only. This owes to the limitation of the linear drives' moving ranges. Thus, the characteristics of Cartesian workspaces can conveniently be displayed in cross-sections of the workspaces parallel to the robot's y-z-plane. See Figs. 4, 6 and 8.

3.2 Workspace Enlargement

The two largest configuration workspaces of the TRIGLIDE-structure are shown in Fig. 3. Corresponding to the PARAPLACER they are denoted as working configurations. Following the above mentioned notation, the two working-configurations are characterized by the configuration vectors $[-1, -1, -1, +1]$ and $[+1, +1, +1, -1]$. As shown in [13], it is again possible to combine both working configurations to an overall workspace which is significantly larger than a single configuration workspace, see Fig. 3. A possible path connecting both working-configurations which leads through three transition configurations is depicted in Fig. 4. It comprises the following four steps:

1. **Working mode change**
   In the first step of the configuration change procedure a singularity of type 1 is passed in point A so that a working mode change is performed. This leads the manipulator from configuration $[-1, -1, -1, +1]$ to configuration $[+1, -1, -1, +1]$.

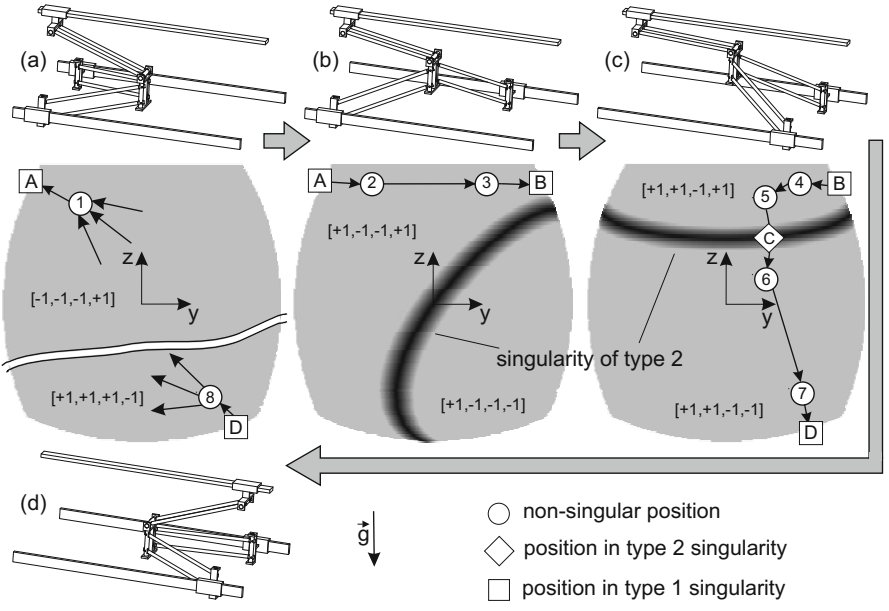


Fig. 4. Possible configuration change sequence for the TRIGLIDE-robot.

2. **Working mode change**
   The second step consists in another working mode change. At this stage the second kinematic chain passes a singularity of type 1 in point B so that the new robot configuration becomes $[+1, +1, -1, +1]$.
3. **Assembly mode change**
   The third step of the configuration change sequence comprises an assembly mode change by which the singularity of type 2 is passed in point C. Afterwards the structure is in configuration $[+1, +1, -1, -1]$. In order to avoid control problems during this step the structure is again underactuated by releasing one of the actuators and a driving force is used to guide the manipulator through the singularity. Whereas inertia is used as the driving force for the PARAPLACER-robot, gravity forces are used here. This is even simpler, because the structure does not need to be accelerated towards the singularity, but just falls under gravity influence if one motor is released. The procedure is described in more detail in [14, 15].
4. **Working mode change**
   Finally, in a fourth step a last working mode change is performed leading the TRIGLIDE-structure from configuration $[+1, +1, -1, -1]$ to the desired stop-configuration $[+1, +1, +1, -1]$. The approach to conduct this working mode change works completely analogous to the other working mode changes.

4 Planning a Configuration Change

As shown above by two examples, it may happen that one or several transition configurations need to be passed in order to conduct a change between a start and a stop configuration. A complete sequence leading from a start configuration to a stop configuration is called a path in the remainder of this article. Usually, different paths exist between two configurations. However, a lot of kinematic knowledge is required in order to find such a path. Furthermore, the most intuitive path might not be the most suitable one. Especially it is desirable to find a path, that minimizes execution time. Thus, a systematic approach is required in order to plan a path between the start and stop configurations. For this reason a powerful approach has been developed which is based on graph theory [14]. It is elucidated next using the TRIGLIDE-robot as example.

The first and most important step in this approach is to set up a so-called configuration-graph. Such a graph is shown in Fig. 5 for the TRIGLIDE-robot. Each node of the graph corresponds to one of the robot specific configurations, described by the vector **k**. Furthermore, each edge of the configuration-graph can be interpreted as the possibility to perform a direct change between the two configurations, which are connected by the edge. The condition for connecting two configurations by an edge is the existence of at least one point in both configuration workspaces for which the end-effector-coordinates as well as the actuator-coordinates of the robot correspond to each other. This condition can only be fulfilled in singularities. Different types of edges exist, as shown in the graph, and, in consequence, different types of singularities. Besides singularities of type 1 and type 2, introduced in section 1, singularities of type 3 exist, in which type 1 and type 2 singularities coincide in one point. Additionally several singularities of the same type may coincide in one point.

The condition, which is responsible for having a connecting edge or not between two configurations, can be automatically checked by workspace discretization under certain conditions [14]. Most importantly, the kinematic equations of the robot under investigation, or more specifically the solutions of both the IKP as well as the DKP, have to be expressible in terms of the configuration-parameters **k**. If this is the case, the configuration-graph can be automatically derived [14]. A further requirement for finding the most suitable path is, that the edges of the configuration-graph are weighted. Whenever a path has to be found, which minimizes the configuration-change duration, the weights correspond to the time required to pass the singularity. However, the weighting is quite complex and not always possible on a high planning level, because this time for passing the singularity depends on a number of robot specific parameters, such as masses, control-parameters, etc.. Hence, in a first attempt it is proposed to impose the same weight on all edges. This simplification proved to be feasible, because it results in a path with the minimal number of path segments. The path with the fewest number of segments is approximately the one which takes the least time, since each path segment amounts to a specific duration.

Once the configuration-graph has been derived, the path planning between any two configurations is possible. For this purpose Dijkstra's algorithm [16] is applied,

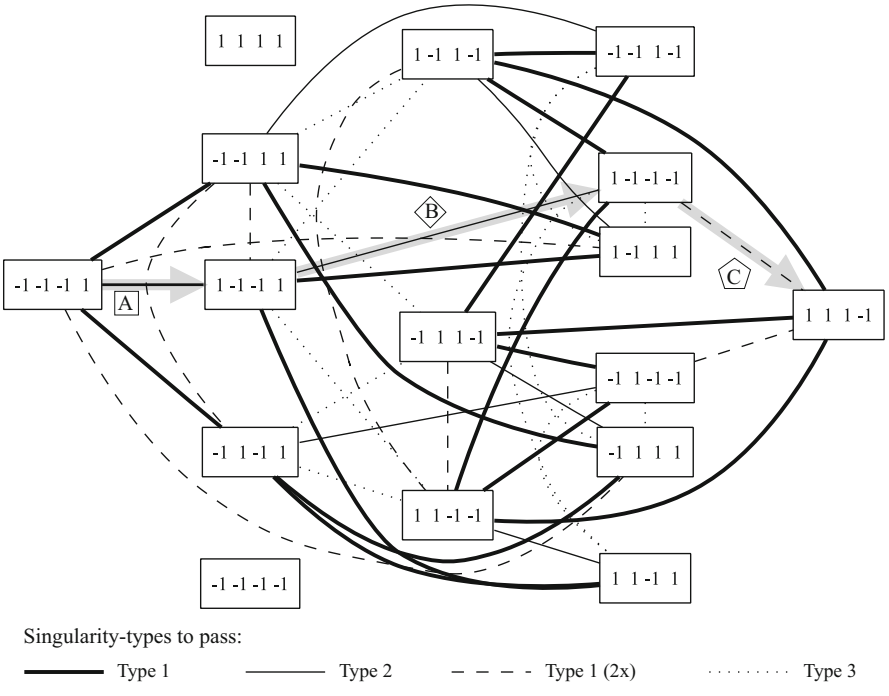


Fig. 5. Configuration graph of the TRIGLIDE-robot and one possible path between the two working configurations.

which allows to find a shortest path through the graph. Here, the term shortest means that the sum of the weights imposed to the edges, which are included in a possible path, is minimal.

It is possible to find a shortest path between two arbitrary configurations in a systematic manner by means of the presented approach. The application to the TRIGLIDE-robot shows, that several paths with shortest length exist for a change between the two working configurations. It is important to mention, that each of these paths with shortest length leads either through a point where two singularities of type 1 coincide or through a singularity of type 3 where type 1 and type 2 singularities coincide. One of these paths is marked in Fig. 5 by the bold gray arrows. Fig. 6 illustrates the corresponding end-effector-movement in cartesian space representation within the individual configuration workspaces. Compared with the four-step path presented in section 3.2 (Fig. 4), this path consists of only three steps:

1. **Working mode change**
   Firstly, a working mode change is conducted by passing a type 1 singularity in point A leading the manipulator from configuration $[-1, -1, -1, +1]$ to configuration $[+1, -1, -1, +1]$. This step exactly corresponds to the first step of the procedure described in section 3.2.

2. **Assembly mode change**
   An assembly mode change is performed in the second step of the procedure. Thereby the configuration changes from $[+1, -1, -1, +1]$ to $[+1, -1, -1, -1]$. The general approach to pass the singularity of type 2 in point B is similar to that in the third step of the four-step configuration change procedure presented above. However, comparing Fig. 4 and Fig. 6 the singularity is passed in another robot-configuration and at a different position.
3. **Working mode change**
   Finally an additional working mode change is peformed in order to reach the desired stop configuration $[+1, +1, +1, -1]$. The essential point is that the configuration is changed in point C in which two singularities of type 1 coincide in a single point, so that two of the configuration parameters included in vector **k** are altered at the same time. This step does not cause any difficulties if the robot is controlled in joint-space, similar to a working mode change by which a single type 1 singularity is passed.

In practice it has been proven that the path execution time is approximately 2.4 s which is significantly shorter than execution times required for alternative configuration change paths [17]. For example, the time needed to conduct the change between the same configurations but following the four-step path given by Fig. 4 is approximately 3.0 s.

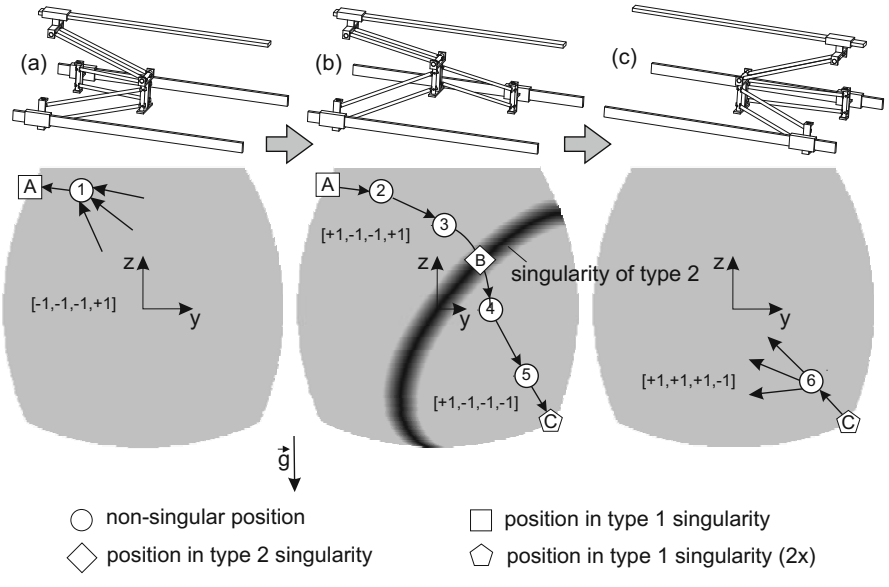


Fig. 6. Possible path with shortest length to follow in order to accomplish a configuration change between the two working-configurations of the TRIGLIDE-robot.

5 Automatic Configuration Detection

If a robot is allowed to work in different configurations the control system must be aware of the current configuration because the solutions of the kinematic and dynamic model equations are dependent on the configuration parameters **k**. Unfortunately, the configuration **k** is unknown directly after starting the robot-control-system. Thus, a determination of the actual assembly mode as well as the working mode is necessary at least once after startup of a robot's control. Manually, this can be performed by moving the manipulator into a certain configuration before starting the control. However, this requires manually movable structures. Since user interaction is needed, this method is not safe against operating errors. Another possibility is to save the last known configuration and to reload this configuration into the control-memory after startup. In this case there is a risk, that the structure is manually moved in the off-state. A wrong input can lead to severe damage, since the configuration is needed not only for control of the structure's movements, but also for many safety functions, such as detection of workspace boundaries due to singularities or detection of self-collisions. Thus, a method for automatic detection of a structure's configuration is required.

As described in [18] various approaches for automatic configuration detection based on different principles exist. They can be categorized by two criteria, namely the employed state of motion of the kinematic structure and by the necessity of additional sensors beside the ones used in the actuators. Obviously, to keep additional costs for the configuration detection low, it is favorable not to be dependent on additional sensors. Moreover, for safety reasons it is preferable, that the robot's structure does not have to be moved in order to determine the configuration, because additional effort is necessary to avoid collisions during these movements. A technique is presented next for configuration detection, which is based on gravity effects. It is validated for the TRIGLIDE-robot. Neither the structure needs to be moved nor extra sensors are required. However, it cannot be applied to the Paraplacer robot, since this technique relies on gravity effects .

The proposed method starts from the assumption that the drive positions $\tilde{\mathbf{q}} = [\tilde{q_1}, \tilde{q_2}, \tilde{q_3}]^T$ of the TRIGLIDE-robot are known from the position measurement systems of the robot. Based on this assumption the configuration detection is split into two parts: First the assembly mode is evaluated so that k_{DKP} is known and after that the working mode can be determined so that the parameters $k_{IKP,i}$ are determined for $i = 1, 2, 3$.

5.1 Assembly-Mode-Detection

The fundamental idea of the approach for assembly-mode-detection consists of the use of the static holding forces required to hold the manipulator at a constant position [19, 18]. It can be used, since the influence of gravity on the spatial TRIGLIDE-structure requires drive forces for static holding of the structure. Given that direct electric drives with no additional gears are used in the design, a direct feedback from gravitational forces on the structure on the drive forces can be expected. The

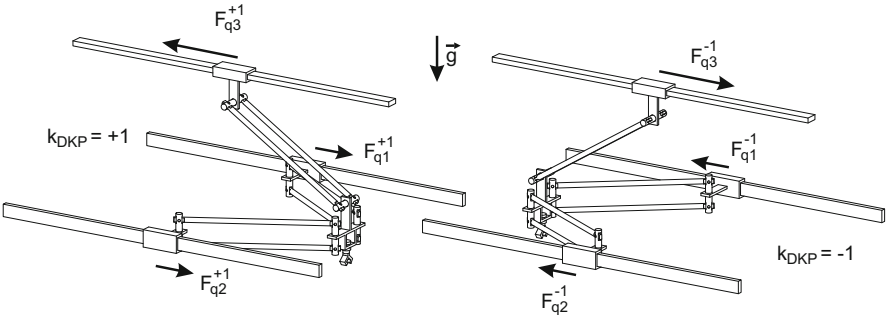


Fig. 7. Drive forces to hold the TRIGLIDE-structure against gravity for both possible assembly modes.

amplifiers for these drives measure the drive currents which are proportional to the forces. Thus, the drive forces $\tilde{F}_i$ for the $i = 1, 2, 3$ actuators can be obtained. Furthermore, it is possible to compute the corresponding necessary theoretical drive forces by means of a dynamic model. In this context it is important to mention that the model equations are a function of the assembly mode parameter k_{DKP}. This means that for a given movement of the drives (position, velocity, acceleration) the dynamic model delivers two possible drive forces F_i^{+1} for $k_{DKP} = +1$ and F_i^{-1} for $k_{DKP} = -1$ for each of the $i = 1, 2, 3$ actuators. This situation is illustrated in Fig. 7.

The basic idea for automatic configuration detection is to compare the computation results F_i^{+1} and F_i^{-1} with the measured forces $\tilde{F}_i$ for the $i = 1, 2, 3$ motors. Under the assumption that the model as well as the measurements are accurate enough (see [14] for necessary examinations), the following distinction is used for identification of the assembly mode parameter k_{DKP}:

$$k_{DKP} = \begin{cases} +1 & \text{if } \sum_{i=1}^{3} \left| F_i^{+1} - \tilde{F}_i \right| < \sum_{i=1}^{3} \left| F_i^{-1} - \tilde{F}_i \right| \\ -1 & \text{if } \sum_{i=1}^{3} \left| F_i^{-1} - \tilde{F}_i \right| < \sum_{i=1}^{3} \left| F_i^{+1} - \tilde{F}_i \right| \end{cases} \tag{1}$$

5.2 Working-Mode-Determination

Once the assembly-mode has been successfully detected the working modes can easily be identified in a similar manner. The first step is to calculate the actual end-effector position by solving the DKP for the actual drive positions $\tilde{q}$ under the consideration of the known assembly-mode-parameter k_{DKP}. This leads to a unique solution. Subsequently the IKP is solved for the $i = 1, 2, 3$ drives. Focussing on the ith drive, the IKP yields two solutions q_i^{+1} and q_i^{-1} for the corresponding working-mode-parameter $k_{IKP,i} = +1$ and $k_{IKP,i} = -1$, respectively. Based on these solutions, the following case-differentiation allows to determine $k_{IKP,i}$

$$k_{IKP,i} = \begin{cases} +1 & \text{if } \left| q_i^{+1} - \tilde{q}_i \right| < \left| q_i^{-1} - \tilde{q}_i \right| \\ -1 & \text{if } \left| q_i^{-1} - \tilde{q}_i \right| < \left| q_i^{+1} - \tilde{q}_i \right| \end{cases} \tag{2}$$

This procedure is repeated for all $i = 1, 2, 3$ drives so that the single elements of the working-mode-configuration-vector $\mathbf{k}_{IKP}$ become available.

5.3 Experimental Experience

The actual working- and assembly-modes are known after successful application of the aforementioned procedures. Thus, all kinematic and dynamic models of the robot can be solved unambiguously, making a safe robot operation possible. Experiments show, that the proposed technique for automatic assembly-mode-detection works well for the TRIGLIDE-robot [14]. Only in small workspace sections, which can be detected, the method fails, because the difference between computed and measured forces is not large enough in order to compensate for measurement and model errors. Fig. 8 shows the areas in which the method does not yield a unique solution (due to symmetry-reasons the three pictures are representative for all other configuration workspaces). A failure is detected if the automatic configuration detection is in such a position. The user is informed and advised to move the structure to a different position before the process is restarted.

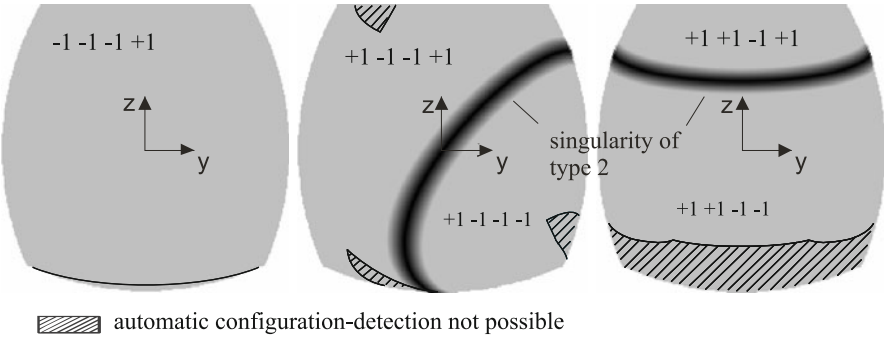


Fig. 8. Workspace areas of the TRIGLIDE-structure, in which automatic configuration detection is not applicable.

6 Conclusion

Changing the configuration of a parallel robot is a measure to enhance its workspace and thereby to overcome one of the major drawbacks to parallel robots. The approach is elucidated by two manipulators - a planar and a spatial example. The workspace is almost doubled for both mechanisms, emphasizing the promising potential of the method. A systematic scheme based on graph theory is introduced in order to simplify the planning process of the configuration-change-procedure. This technique is not only capable to find all possible paths between the configurations which are supposed to be connected. Especially it permits the detection of the shortest path, making the process as efficient as possible. Finally, an automatic configuration-detection-strategy is presented which allows for safe robot operation.

Such a technique is required because all the model equations considered in the controller are dependent on the configuration parameters. Hence, this method is applied directly after start-up of the robot. In conclusion, the promising potential of the configuration change technique is validated. Moreover, it is complemented and optimized in terms of efficiency and safety by featuring special supporting functionalities. It therefore suits industrial needs.

Acknowledgements. This work was funded by the German Research Foundation (DFG) within the framework of the Collaborative Research Center SFB 562: Robotic Systems for Handling and Assembly.

References

1. Helm, M.: Durchschlagende Mechanismen für Parallelroboter. In: Schriftenreihe des Instituts für Werkzeugmaschinen und Fertigungstechnik der TU Braunschweig. Vulkan-Verlag, Essen (2003) ISBN 3-8027-8671-8
2. Merlet, J.-P.: Parallel Robots. Springer, Dordrecht (2006)
3. Wenger, P., Chablat, D.: Workspace and assembly modes in fully-parallel manipulators: A descriptive study. In: Proceedings of the International Symposium on Advances in Robot Kinematics, Salzburg, Austria, pp. 117–126 (1998)
4. Gosselin, C., Angeles, J.: Singularity analysis of closed-loop kinematic chains. IEEE Trans. on Robotics and Automation 3(6), 229–281 (1990)
5. Bonev, I.A., Chablat, D., Wenger, P.: Working and assembly modes of the agile eye. In: Proceedings of the IEEE International Conference on Robotics and Automation, Orlando, USA, pp. 2317–2322 (2006)
6. Hovland, G., Choux, M., Murray, M., Tyapin, I., Brogardh, T.: The gantry-tau - summary of latest development at ABB. In: Proceedings of the 3rd International Colloquium of the Collaborative Research Center 562, Braunschweig, Germany, pp. 141–158 (2003)
7. Hesselbach, J., Helm, M., Soetebier, S.: Connecting assembly modes for workspace enlargement. In: Lenarcic, J. (ed.) Advances in Robot Kinematics. Kluwer Academic Publishers, Dordrecht (2002)
8. Hesselbach, J., Helm, M., Kunzmann, H.: Workspace enlargement for parallel kinematic machines. CIRP Annals 52(1), 343–346 (2003)
9. Schütz, D., Budde, C., Raatz, A., Hesselbach, J.: Parallel Kinematic Structures of the SFB 562. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 109–124. Springer, Heidelberg (2010)
10. Dietrich, F., Maaß, J., Bier, C., Pietsch, I., Raatz, A., Hesselbach, J.: Detection and Avoidance of Singularities in Parallel Kinematic Machines. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 77–92. Springer, Heidelberg (2010)
11. Hesselbach, J., Helm, M., Soetebier, S., Rathmann, S.: Singularity passing with parallel robots. In: Modern Technologies in Manufacturing, Cluj-Napoca, Romania, pp. 245–248 (2003)
12. Budde, C., Last, P., Hesselbach, J.: Development of a Triglide-robot with enlarged workspace. In: Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy, pp. 543–548 (2007)

13. Budde, C., Last, P., Hesselbach, J.: Workspace enlargement of a Triglide robot by changing working and assembly mode. In: IASTED International Conference on Robotics and Applications, Cambridge, USA, pp. 244–248 (2005)
14. Budde, C.: Wechsel der Konfiguration zur Arbeitsraumvergrößerung bei Parallelrobotern. In: Schriftenreihe des Instituts für Werkzeugmaschinen und Fertigungstechnik der TU Braunschweig, Vulkan-Verlag, Essen (2009)
15. Maaß, J., Kolbus, M., Budde, C., Hesselbach, J., Schumacher, W.: Control strategies for enlarging a spatial parallel robot's workspace by change of configuration. In: Proceedings of the 5th Chemnitz Parallel Kinematics Seminar, Chemnitz, Germany, pp. 515–530 (2006)
16. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
17. Budde, C., Kolbus, M., Last, P., Raatz, A., Hesselbach, J., Schumacher, W.: Optimized change of working and assemby mode of the SFB 562 triglide-robot. In: Proceedings of the 3rd International Colloquium of the SFB 562: Robotic Systems for Handling and Assembly, Braunschweig, Germany, pp. 221–236 (2008)
18. Budde, C., Rose, M., Maaß, J., Raatz, A.: Automatic detection of assembly mode for a Triglide-robot. In: Proceedings of the IEEE International Conference on Robotics and Automation, Pasadena, USA (2008)
19. Budde, C., Maaß, J., Raatz, A., Hesselbach, J.: Automatisierte Konfigurationserkennung bei einem Triglide-Roboter. In: Proceedings of the Robotik, Munich, Germany (2008)

Part III
Control and Programming

A Middleware for High-Speed Distributed Real-Time Robotic Applications

Bernd Finkemeyer, Torsten Kröger, and Friedrich M. Wahl

Abstract. The development of modular and distributed real-time software systems — as they are common in the field of research and development of robotic manipulation control systems — can be greatly simplified by appropriate middleware concepts. This paper generically introduces the basic concepts of the middleware solution MiRPA (*Middleware for Robotic and Process Control Applications*). MiRPA's employment allows the implementation of complex distributed real-time software architectures. It handles publisher/subscriber as well as client/server communication between local and distributed software modules with very small worst-case latencies ($\ll 1\,ms$). Besides introducing basics on inter-module and inter-node communication for these two communication models, deadlock avoiding strategies, and methods of redundancy handling, we finally show results of real-world measurements and applications.

1 Introduction

The complexity of software architectures in research and development projects of robot control systems is continuously increasing. In order to obtain modular, extensible, and flexible software structures, middleware-based approaches have proven

Bernd Finkemeyer
KUKA Roboter GmbH, Zugspitzstraße 140, 86165 Augsburg, Germany
e-mail: berndfinkemeyer@kuka-roboter.de

Torsten Kröger
Stanford University, Artificial Intelligence Laboratory, Department of Computer Science, Stanford, CA 94305-9010, USA
e-mail: tkr@stanford.edu

Friedrich M. Wahl
Technische Universität Braunschweig, Institute for Robotics and Process Control, Mühlenpfordtstraße 23, 38106 Braunschweig, Germany
e-mail: f.wahl@tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 193–212.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

to be of high practical relevance: Huge numbers of commercial and/or freely available solutions are available for many applications — including robotic applications. In particular the field of mobile robotics features many software frameworks (e.g., [1, 2, 3]). Nevertheless, in the field of robotic manipulation control, the additional necessity of low worst-case latencies for the intercommunication among software modules, i.e., *real-time capability* is required, as control algorithms have to be executed in isochronous time intervals. Before introducing further related works in Sec. 2, we clarify the focus of this paper, i.e., we generally describe the basic concepts for realizing the thin, high-performance real-time middleware solution MiRPA (*Middleware for Robotic and Process Control Applications*).

1. *Client/server communication.* As most real-time and non-real-time software systems can be divided into several modules with client or server functionality, a real-time middleware has to support this communication model. Furthermore, it has to be possible to use synchronous and/or asynchronous server requests.
2. *Publisher/subscriber communication.* Software modules have to be able to subscribe messages from other modules. This way, (sporadically and/or isochronously generated) information can be easily distributed within the entire system.
3. *Ad hoc and context sensitive communication.* Several applications as well as many use-cases during software development processes require (very) flexible communication structures, such that communication partners may depend on their current contexts. A manufacturing machine, for example, may be provided with materials by different mobile robots, i.e., the communication partners of the machine change frequently.
4. *On-line exchange of software modules.* Enhancing software systems during their development process commonly requires a stopping of the affected application. In order to provide the possibility of frequent software changes in concrete software components, these modules have to be exchangeable during runtime. Furthermore, if a middleware is used in a non-development environment, this feature can be used for on-line software updates without shutting down the complete system.
5. *Fault-tolerance.* Due to the fact that the client/server communication model *and* the publisher/subscriber communication model are supposed to be implemented, the occurrence of cyclic deadlocks is principally possible. In order to prevent the system from being unavailable, such deadlocks have to be actively recognized by the middleware instance.
6. *Routing.* Complex applications often contain several different physical and/or logical networks — including interfaces and/or networks of non-real-time environments. The middleware (running on all nodes of the entire network) has to be responsible for the routing functionalities within this network.
7. *Portability.* As mentioned in the previous item, the overall system, on which the middleware runs, may be very heterogenous, such that, for example, different operating systems are used on different nodes. In order to have a unified interface to the middleware, it is required to have implementations on several hard- and software platforms.

8. *Limited worst-case latencies.* This is the most-relevant requirement: Users have to be able to determine a worst-case latency time that is required for the exchange of data between two software modules running on the same or on different computing nodes. For several mechatronic (and in particular, robotic) applications, worst-case latencies of (much) less than $200\,\mu s$ are desired.

In particular the last item features the major difference of MiRPA in comparison to other solutions known up to now. Nevertheless, all the listed properties must be fulfilled by a middleware suited for real-time applications in the field of robotics.

After the introduction of related works in Sec. 2, Sec. 3 — the core part of this contribution – introduces the basic concepts of the technologies used for the realization of MiRPA. Real-world time and latency measurements are shown in Sec. 4, while Sec. 5 illustrates a demanding sample application in the field robot motion control.

2 Related Work

Basics and a good middleware overviews are presented in [4, 5, 6]. Fundamentals of middleware usage in real-time and embedded systems are outlined in [7].

CORBA (*Common Object Request Broker Architecture*) is the most common middleware; it has been specified by the OMG (*Object Management Group*, [8]). CORBA allows object oriented client/server interactions in a transparent way; several programming languages and operating systems are supported. It is kept very open and some of the properties mentioned in Sec. 1 have been specified as CORBA services by the OMG. In order to realize publisher/subscriber communication, the CORBA Event Service can be used. The idea of fault tolerance is considered in the FT-CORBA specification. Nevertheless, the achieved QoS (quality of service) is not suited for real-time demands, as no worst-case latencies can be guaranteed. To overcome this problem, the OMG founded a SIG (*special interest group*) to prepare a real-time ORB (*object request broker*) specification [8] in 1996. A survey of research towards real-time CORBA and methods enhancing the QoS are presented in [9] and [10].

ACE TAO [11, 12, 13], ORBexpress [14], and OpenRTM-aist [15] are well known RT-CORBA implementations. Priority scheduling algorithms and optimized communication protocols have significantly increased the QoS, which has been verified in numerous publications (e.g., [16, 13, 17]). But these improvements are not far-reaching enough for many real-time applications, like for low-level controllers with cycle times of less than one millisecond and synchronization requirements of values in the range one microsecond. The performance is additionally limited by the above mentioned CORBA services, which are very important for many real-time applications.

In [18, 19], an approach for the integration of real-time programming and CORBA is presented. Real-time and non-real-time software parts are connected via so-called *Composite Objects* to a single object-based framework, where the distributed real-time part is based on the Simplex Architecture [20], i.e., a real-time

publisher/subscriber communication model [21]. The Simplex Architecture additionally enables on-line software upgrades [22] — a problem, which is widely discussed in the open real-time literature (e.g., [23]).

The publisher/subscriber communication model has enormous relevance in the field of real-time applications. It has already been mentioned that CORBA emulates this communication model with the CORBA Event Service, which constitutes an additional CORBA server handling publisher/subscriber communications. This indirect way is necessary, as CORBA's communication protocol is based on a client/server model. For real-time applications this solution is not efficient enough. To overcome this lack, the OMG has specified the middleware DDS (*Data Distribution Service for Real-Time Systems*) [8]. It is based on the commercial publisher/subscriber middleware, whose former name was NDDS (*Network Middleware for Distributed Real-Time Applications*). Nowadays, it is entitled *RTI Data Distribution Service* [24]. This middleware also offers an API (application programming interface) to emulate client/server communication.

Many middleware approaches are of very static nature. They assume that the relationship between all involved software modules is fixed at compile time. Existing dynamic approaches, for example, dynamic CORBA, scarcely fulfill real-time requirements. In contrast to this, very flexible and adaptable control systems are required by many automatization applications. Numerous special approaches or CORBA extensions for adaptive and dynamic middlewares (e.g., [25, 26, 27]) or ad hoc networks in mobile computing (e.g., [28, 29, 30]) can be found in the literature. These approaches scarcely fulfill all the remaining properties of a middleware for process control applications, which have been listed in Sec. 1. It is well known, that appropriate software development environments for robotic applications are indispensable. One recent approach by the German DLR Institute of Robotics and Mechatronics is called aRD (*agile Robot Development*). In [31], an overview of the aRD software architecture is given. OROCOS (*Open Robot Control Software*, [32, 33]) also applies middleware as well as *Orca*, a CBSE (*Component-Based Software Engineering*) framework for mobile robotics [2, 34], Player [1, 35], the Microsoft Robotics Studio [36], CLARAty [37], and Miro (*Middleware for Mobile Robot Applications*, [38]) do.

We may summarize, that common middleware solutions only cover an individual subset of the required characteristics of a middleware for high-performance process control applications. Especially the combination of high QoS, which enables distributed feedback control of highly dynamic plants, and services like active deadlock avoidance, fault tolerance, administration of redundancies, and easy programmability, for example, by rapid control prototyping tools, is still not supported in a sufficient way. Thus, further work in the field of real-time middleware is necessary. As a contribution to this middleware discussion, our middleware solution MiRPA (*Middleware for Robotic and Process Control Applications*) is presented in this paper; preliminary papers on this subject are [39, 40, 41]. A highly specialized variant, *MiRPA-X* (MiRPA *Extended*), has been developed [42, 43, 44] and has been applied in the SFB 562 for high-performance control applications of parallel kinematic structures [43, 45].

Table 1. Structure of a MiRPA message object.

4 byte	Type			
4 byte	Length			
4 byte	Priority			
4 byte	Time limit			
n byte	Name			
m byte	Parameter list	Parameter_1	1 byte	Data Type
			4 byte	Length
			p_1 byte	Data
		...	...	...

3 MiRPA Functionality

MiRPA has been designed for the requirements of real-time systems as pointed out by the eight items listed in the introductory section. This section briefly introduces the most relevant key concepts used for the realization of MiRPA.

3.1 Message Objects

MiRPA is a message-based middleware, i.e., all software modules connected to MiRPA communicate via message objects. Such objects contain user data and protocol data as well; Table 1 shows the structure of the MiRPA communication object.

The four bytes of the field *Type* determine the function of the message and are part of the MiRPA protocol. Table 2 gives an overview of available message types.

The length of a message object is given by the part *Length* in bytes. It is possible to specify a normalized priority ranging from 0 (lowest priority) to 100 (highest priority) for each message within the section *Priority*; the message priority is responsible for the real-time capability of MiRPA (cf. Sec. 3.4). The maximum time for the message transfer can be explicitly specified by the *Time limit* field. Each message is unambiguously denoted by the *Name* field. The *Parameter list* contains the user data of the message; it may be any kind of complex data structure.

Table 2. Message types used by MiRPA.

Message Type	Functionality
`GENERAL`	This message is delivered to one or more subscribers.
`REQUEST`	Asynchronous server request, which will be delivered to one or more servers.
`REQUEST_BLOCK`	Synchronous server request, which will be delivered to one or more servers.
`ANSWER`	Answer to a request.
`GET_ANSWER`	Gets the answer of an asynchronous request.
`POLL_ANSWER`	Checks if the answer of an asynchronous request is available.
`WAIT`	Indicates that an answer is not available yet.
`CONTROL`	Activates MiRPA services.
`MSGOK`	Acknowledges the successful forwarding of a message.
`MSGUNKN`	Informs the sender that the subscriber or server is unknown.
`INVALID`	Informs the sender that an error has occurred.

3.2 Communication

MiRPA supports client/server as well as publisher/subscriber communication. For client/server communication, a synchronous and an asynchronous alternative mode can be used. These different possibilities are achieved by MiRPA's general form of forwarding the above mentioned communication objects. The fundamental principle is always the same: Software modules register the names of messages, which they want to receive. This registration is done with the MiRPA service named `logon`. The service is activated by a message of the type `CONTROL`. Its parameter contains the *Name* of the message to be registered. MiRPA administers all registered names and software modules in a list. When MiRPA receives a message, it checks its name and forwards the message to all software modules, which have registered this name. To which communication model the message belongs, or how the message must be interpreted, is determined by the field *Type* of the message. The different alternatives will be described below.

3.2.1 Publisher/Subscriber Communication

This communication model is outlined in Fig. 1: The two software modules, `Subscriber1` and `Subscriber2` register the message named `EventA` with a MiRPA `CONTROL` message named `logon`, which is acknowledged with a `MSGOK` message by MiRPA. This way, both modules subscribe the message `EventA`. From now on, if any module sends a message named `EventA`, it will be forwarded to the mentioned subscribers. Publisher modules sending a `GENERAL` message receive a `MSGOK` message containing the number of subscribers that have received the message within the given time interval determined by the publisher in the *Time limit* field of the `GENERAL` message. If MiRPA is not able to forward the message, a `MSGUNKN` message is returned to its sender. It should be remarked, that every software module can work as a publisher as well as a subscriber, i.e., also the processes

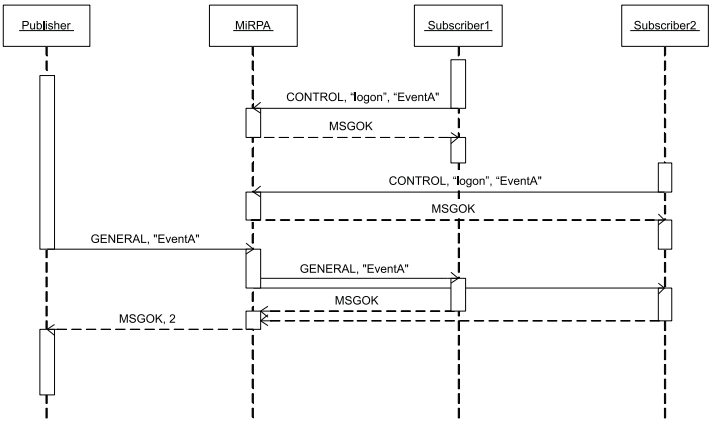


Fig. 1. Sequence diagram of MiRPA's publischer/subscriber communication.

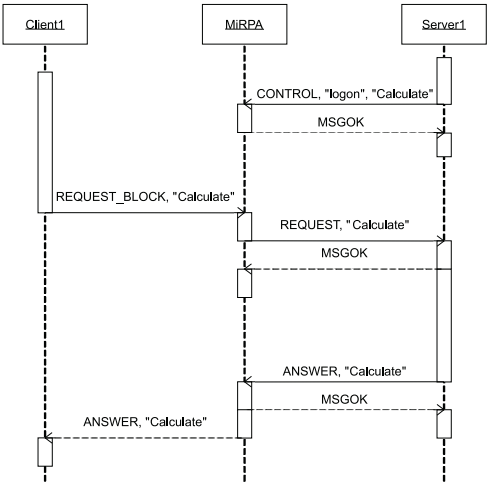


Fig. 2. Sequence diagram of MiRPA's synchronous server access.

`Subscriber1` and `Subscriber2` of the example in Fig. 1 can send the message `EventA` (i.e., many to many communications are possible).

3.2.2 Synchronous and Asynchronous Client/Server Communication

This communication model allows the direct access to server modules, whereas MiRPA distinguishes between synchronous and asynchronous server requests. In the case of synchronous requests, the client is blocked until the server has answered; this blocking can be avoided by using the asynchronous alternative. Here, answers of server modules are explicitly fetched by the client at an arbitrary point of time and without blocking. The UML sequence diagram of the synchronous server request is outlined in Fig. 2. In the first step, the server registers the service named `Calculate` at MiRPA. This procedure is similar to the subscriber variant (cf. Sec. 3.2.1, Fig. 1). Now, any client process is able to use this service. As one can see in the sequence diagram of Fig. 2, the module `Client1` sends a `REQUEST_BLOCK` message, which induces a synchronous request, to MiRPA. `Client1` will be blocked until an answer is available. After the message is forwarded to one (or more) server(s), the addressed server module(s) acknowledge the reception of the request by a `MSGOK` message. After executing the service, the server module(s) sends its `ANSWER` message to MiRPA that will direct the message to requesting client module.

If the client module sends a synchronous request (i.e., a `REQUEST_BLOCK` message), it blocks and receives only one answer. If there are more than one server modules offering the service to a particular request (i.e., redundant server modules running one or more computing nodes), the client module receives only the answer of the fastest server. In case of asynchronous requests by the same client module (i.e., a `REQUEST` messages), the client can decide to receive either

- the `ANSWER` message of the fastest server module,

- one or more `ANSWER` messages of particular server modules, or
- all available `ANSWER` messages.

The latter case is, for example, important for sensor data fusion applications, but as a general result and depending on the application to be realized, system developers obtain the important ability of setting up their software architecture in a very open, transparent, and flexible way.

3.3 MiRPA Services

MiRPA offers several services to simplify the development of distributed real-time software applications.

3.3.1 Name Service

MiRPA's key function is the real-time name service. In contrast to common middleware approaches, the communication partner is not determined in a separate start-up phase, but is found during runtime. Ad hoc networking and the exchange of software modules during runtime becomes very easy, and a consistent system is guaranteed; software systems can be upgraded without rebooting.

3.3.2 Client Limitation

Software systems based on MiRPA are very flexible and transparent. As mentioned, many to many communications are possible. But in some applications this may lead to dangerous situations: In a control system, for example, it is undesirable that a control algorithm receives reference values from different modules at the same time as the controller would not be able to handle potential contradictions. To avoid this, MiRPA allows clients to reserve services.

3.3.3 Deadlock Avoidance

Especially in real-time applications, the occurrence of deadlocks may have fatal consequences, thus, the avoidance of deadlocks has to be guaranteed. Since in MiRPA-based systems, all module interactions are handled by MiRPA, the entire communication can be observed by the middleware instance running on each node. This facilitates the early detection of potential deadlocks and corrupted modules. MiRPA offers several mechanisms to detect communication problems:

- The complete communication process of each message is observed separately by a watch dog. If a specified time limit is expired, the communication attempt is abandoned by MiRPA and an error message is replied to the sender.
- MiRPA uses a separate thread for the communication with each receiver process, i.e., a server or a subscriber. As a consequence, MiRPA itself can never be blocked and intercommunication processes cannot be affected by any corrupt server or subscriber modules.

- In a MiRPA system, the communication paths are very flexible and may change dynamically. A process may act as a client and as a server at the same time, which leads to the problem of cyclic deadlocks. Existing approaches to avoid such deadlocks are based on algorithms using directed graphs (e.g., a wait-for-graph). If the graph represents a DAG (*directed acyclic graph*), the execution will be free from deadlocks. Numerous algorithms exist to detect a DAG. But in a MiRPA system, the communication pathes and involved modules may rapidly change. As the administration of the graph and the DAG validation is too time-consuming, common algorithms to check for deadlocks are not suited for real-time systems. To cope with the requirements, MiRPA — additionally — uses an internal request list, which contains all requests that have not been answered yet. Before a request is forwarded to a server module, it is verified, that this server does not have any open request in the list.

3.4 Priority-Based Message and Thread Scheduling

In order to be real-time capable, we have to guarantee that the message with the highest priority in the entire system is treated prior to all other messages. This message scheduling is very tightly related to process and thread scheduling — both have to be realized in *one* consistent scheduling framework.

Remark 1. Scheduling framework in this context does not only include the scheduling behavior on one single computing node or on one single communication channel, but the scheduling behavior of the *entire* system. Real-time computing nodes and real-time capable communication channels are commonly not real-time capable if only composed together in one overall system. A system composed of RTnet [46] and Linux Xenomai [47], for example, is not real-time capable as priority inheritance is not possible through the network.

In a MiRPA-based software architecture, the message and the thread scheduling are always sender-driven. The priority of a sender is written to the field *Priority* of the sent message (cf. Table 1). This priority is inherited by the receiving instance (i.e., always MiRPA), such that — if the FIFO scheduling algorithm is applied by the thread/process scheduler — the receiving thread is scheduled next. If MiRPA understands the message (i.e., there is at least one module that registered this message), the priority of the message (i.e., the priority of the sender) will be given to one of the sender threads within MiRPA, and this thread will then forward the message to the actual server or subscriber. If there is more than one recipient module for this message, the respective number of sending threads within MiRPA will be activated analogously. Again the priority will be inherited by the receiving module(s). Depending on the message type (cf. Table 2), these modules may generate an `ANSWER` message that will be send back the original sender analogously. If one of the modules — a thread of MiRPA, a server, or a subscriber module — is busy, because it handles a message, and a higher priority message arrives, the priority of this module increases to the higher one, such that right after the handling of the first message is finished, the high priority message can be treated immediately.

This behavior is not only realized for intercommunications on single nodes but also for intercommunications throughout the network among different nodes. One major requirement for this concept is that priority inheritance can be realized through the network, i.e., if a message arrives from the network, the receiving module (i.e., the MiRPA receiver) inherits the priority of the sending thread on the remote node. This behavior can be easily implemented on micro kernel operating system architectures (e.g., [48]), whereas monolithic operating systems (e.g., [47, 49]) are commonly not suited at all for inter-node communication under real-time conditions, because the problem of priority inversion cannot be solved.

The concept of the sender-driven priority inheritance is furthermore absolutely consistent with the deadlock avoidance strategy (cf. Sec. 3.3.3); as deadlocks are prevented in any case, the problem of priority inversion cannot occur as a matter of principle.

4 Time and Latency Measurements

This section introduces performance measures of middleware solutions in the field of real-time applications and uses MiRPA as sample middleware in order to provide methods for its evaluation as well as for its real-time verification. After describing the experimental set-ups, measures and procedures on time and latency values are presented.

4.1 Experimental Environment

4.1.1 Hardware

Two exacly equivalent PCs are used for the measures presented in this section: AMD Athlon64 3700+ (2.2 GHz, 1024KB L2 Cache), 2GB DDR-400, Gigabyte GA-K8NF9 Ultra F5 Mainboard, Intel PRO/1000MT Gigabit Ethernet network adapter (two nodes directly connected via a cross Ethernet cable). Unnecessary hardware (USB, audio, etc.) has been disabled for all measurements.

4.1.2 Software

In order to make the results of the upcoming subsections repeatable, we specify the used operating system software:

QNX Neutrino™	:	Version 6.3.0 with Service Pack 3 (32-bit)
Linux™**Xenomai**	:	Version 2.3.1, kernel built with SMP workaround, average latency according to xeno-test: 26.819 μs (32-bit)
Linux™	:	openSUSE version 10.1, Linux kernel version 2.6.20.1 (32-bit)
MS Windows™	:	Windows XP with Service Pack 2 (32-bit)

4.2 Time Measurements

The measurements presented in this paragraph are *not* relevant for the real-time capability of a middleware solution; they are given for the comparison between different platforms and to indicate the possible data throughput depending on the message size. All times are averaged and measured via software only, i.e., a client module sends 1000 messages of a given size (10, 50, ..., 5000 bytes) via the middleware to a server module. The answer message of the server contains the same amount of (new) data. A time measurement starts with the sending of the client and ends as soon as the client receives the answer of the server (i.e., the time for the bidirectional communication is measured).

4.2.1 Local Measurements on One Node Only

Local measurements can be performed in two different ways:

1. All threads (incl. MiRPA) work within one single process only, and
2. the client and the server thread are executed within one process, and the middleware instance itself is started as a separated process.

The non-real-time solutions do not use priority inheritance when communicating via synchronous message passing. The two real-time systems were build in two ways:

(a) correct priority inheritance from the client over MiRPA to the server (real-time capable, cf. Sec. 3.4), and
(b) usage of fixed priorities (FP).

The latter method leads to improved best-case latencies, but it is not real-time capable as priority inversions may appear. Fig. 3 shows the measurement results of local bidirectional intercommunications via MiRPA on different operating systems. Only two of the shown variants and protocols are real-time capable, i.e., the communication via local synchronous message passing on the QNX platform (Qnet) and the communication via XIPC (Xenomai inter-process communicaion; an in-house development) on the Linux Xenomai platform. The inverse values of this diagram give an indication of average data throughput values for given message sizes. For example, the Qnet variant running on QNX Neutrino would provide a data throughput of five megabytes per second if request messages (and answers) of 100 bytes are sent permanently.

4.2.2 Time Measurements in a Distributed Software System

Here, we repeat the measurements from the previous paragraph, but during the distributed measurements, two nodes are involved in communication: One node with a server and MiRPA within one process, and another node with a client and MiRPA within one process. During one measurement, only one protocol is used (e.g., client → TCP → MiRPA (instance on node 1) → TCP → MiRPA (instance on node 2) → TCP → server, and back). The results are shown in Fig. 4.

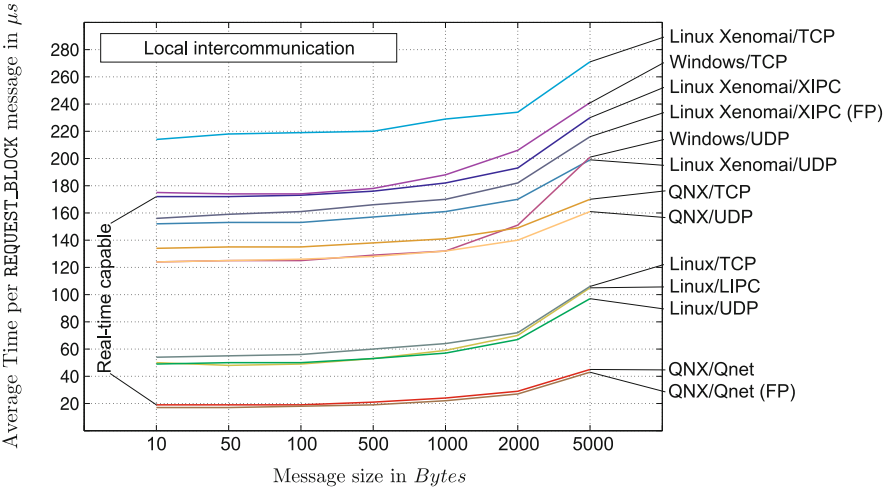


Fig. 3. Averaged time measurements of local bidirectional intercommunications via MiRPA (all threads in one process; not relevant for the real-time capability).

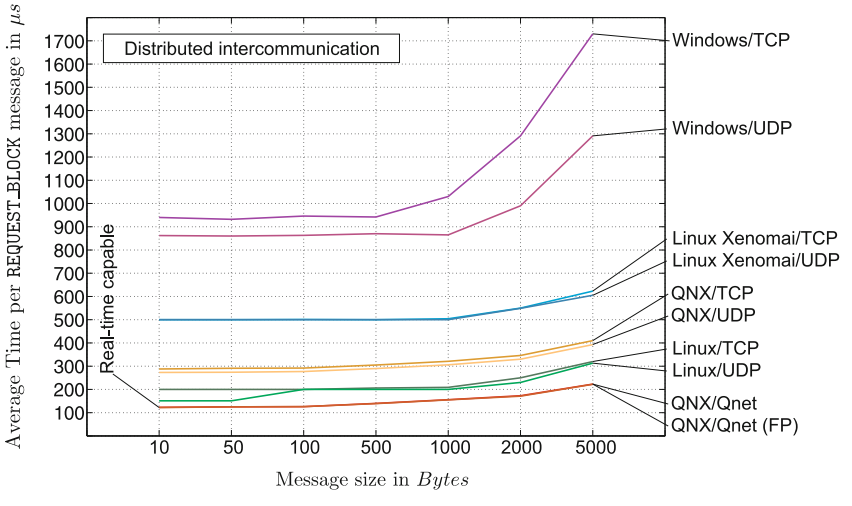


Fig. 4. Averaged time measurements of distributed bidirectional intercommunications via MiRPA (all threads in one process; not relevant for the real-time capability).

Comparing the results of distributed and local experiments leads to an important conclusion: The latency is scarcely influenced by MiRPA. Changing the communication medium has had more influence on the measured times than the usage of the middleware. Thus, the cost-value ratio of MiRPA is very favorable.

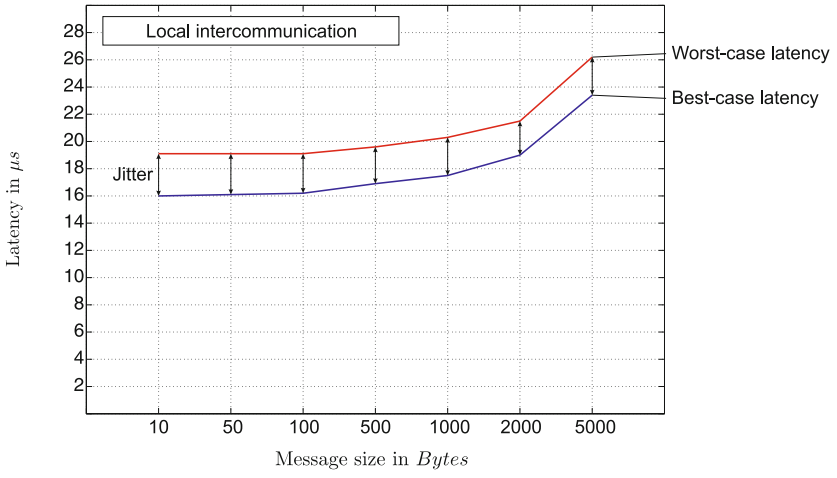


Fig. 5. Latency measurements of local intercommunications via MiRPA using QNX and Qnet (MiRPA as a separated process; relevant for the real-time capability).

4.3 Latency Measurements

Regarding the experimental verification of MiRPA's real-time capability, this part is the most relevant one. All latencies are measured via an oscilloscope. A client module sends one messages of a given size (10, 50, . . . , 5000 bytes) via MiRPA to a server module. The latency and the maximum jitter values are measured, i.e., the

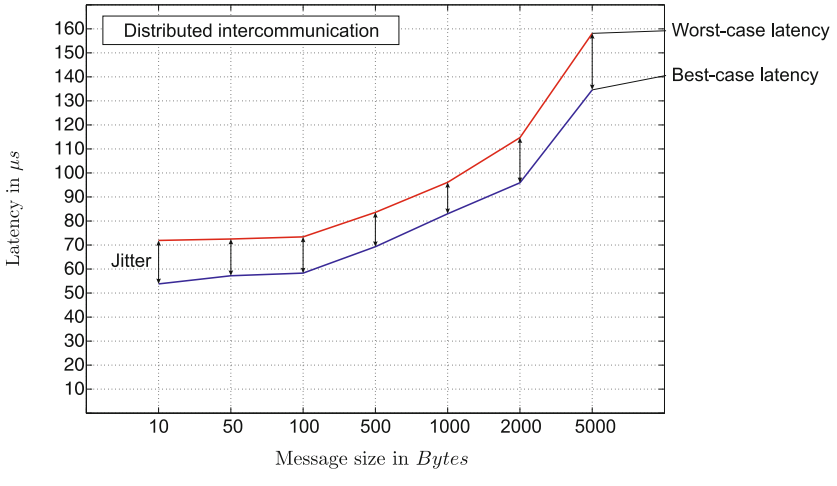


Fig. 6. Latency measurements of distributed intercommunications via MiRPA using QNX and Qnet (MiRPA as a separated process; relevant for the real-time capability).

worst- and the best-case times from the sending action of the client process till the reception of the server process. The local measurements are executed with a client and a server thread working in one process, and MiRPA itself is started within an own process.

Figs. 5 and 6 show the results for the hardware described in Sec. 4.1.1 running QNX as real-time operating system on it. Of course, the exact absolute latency depends on several facts (e.g., the used hardware, the involved operating system, its configuration etc.); Figs. 5 and 6 only indicate the results of one and only one concrete set-up.

5 Sample Application in Robotics

MiRPA is the communication basis for an in-house development of a robot control architecture at the Institut für Robotik und Prozessinformatik at the Technische Universität Carolo-Wilhelmina zu Braunschweig. The software MiRPA-X focusses on different properties and is applied in the Sonderforschungsbereich 562 (Collaborative Research Center; funded by the DFG, German Research Association; cf. [42, 43, 45, 44]). These robot motion control systems are characterized by easy multiple sensor integration, which is achieved by a hybrid control approach (cf. [50]), and high scalability (cf. [51]). Fig. 7 illustrates one possible software architecture of a control system; it will be roughly explained in this section. [40] explains this basic approach in detail; here, we focus only on the usage of MiRPA and its services.

The module *Robot Task* contains the robot program. It sends robot commands, the so-called *Manipulation Primitives* via the *Manipulation Primitive Interface* to the *Manipulation Primitive Execution* module, which presents the core of the hybrid control approach [40, 50]. It enables closed-loop control of any physically measured value for any degree of freedom. The robot may be force controlled in one direction,

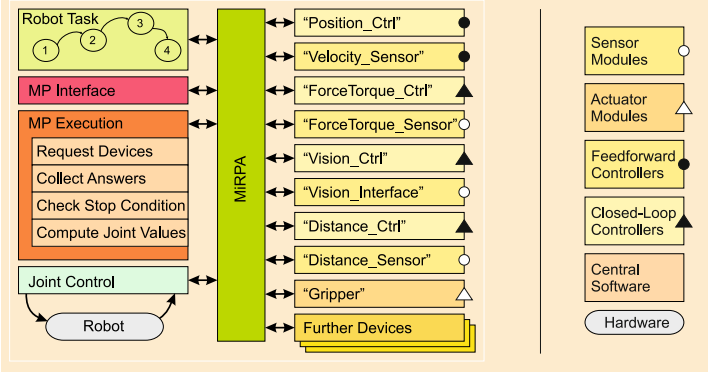


Fig. 7. Software architecture of a robot control system. The whole communication is based on MiRPA (cf. Fig. 5 in [50]).

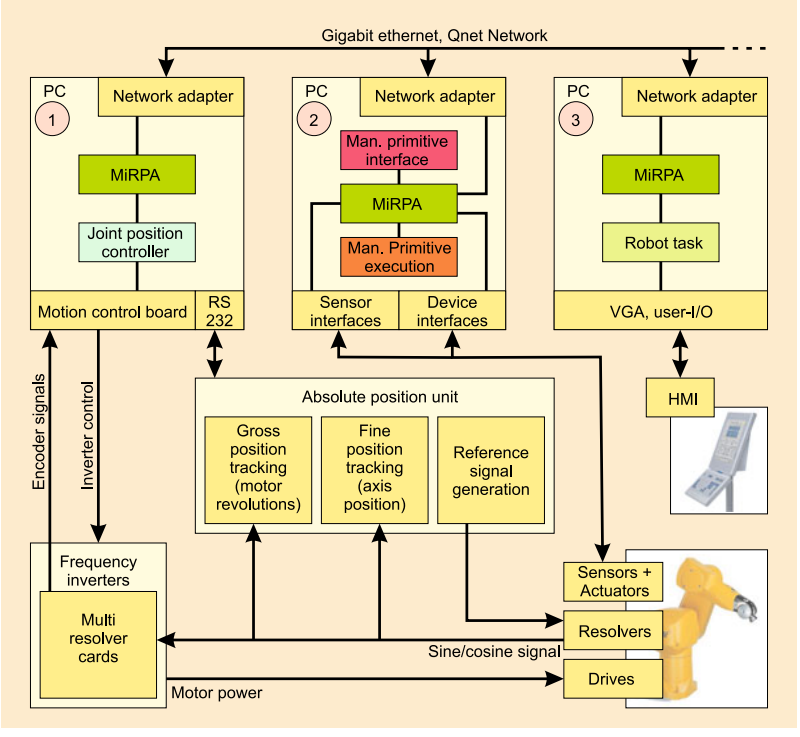


Fig. 8. Hardware architecture of the interfaced Stäubli RX60 industrial manipulator (cf. [52]).

distance controlled in another direction, and pose controlled in a further direction, for example. This hybrid control algorithm is executed in isochronous time intervals. As it is very important to achieve a stable control clock, the *Execution* module runs under hard real-time conditions. Here, the control clock is set to a rate of 1 kHz. In each control cycle, the module triggers the involved controllers and/or sensors, respectively. They are depicted as blocks on the right side in Fig. 7. Here, MiRPA's feature to request modules asynchronously is a big advantage: It enables the *Execution* module to request several control modules, and in the next step, their answers are collected. Thus, if a multiprocessor computer is used or the modules are distributed in a real-time network, they can be executed in parallel.

Fig. 8 depicts the hardware architecture corresponding to Fig. 7. Here, three computer nodes are used to control an Stäubli RX60 industrial manipulator [53].

As MiRPA supports redundant servers — it is very easy to integrate redundant controllers. In this way, the security and robustness of the overall control system can be increased.

The feature of exchanging modules even during runtime is very useful for robot control applications as systems can be upgraded easily. A controller can be started on demand or a stationary robot can establish a connection to a mobile robot to command it or to access its sensors. As there may be numerous mobile robots, the

possibility to build ad hoc connections is essential. During the development process, one can easily exchange controller modules. As also shown in [54], these modules can be directly generated out of Matlab/Simulink [55] during runtime, such that a very convenient development environment exists, and rapid control prototyping can be performed very efficiently.

The redundancy administration of MiRPA and the on-line registration of new modules enable on-line software updates. For example, the update of the module *Force_Ctrl* (cf. Fig. 7) requires only two steps: The newer version must be started. Afterwards, the *Force_Ctrl*'s services are registered two times, and MiRPA will forward messages to the older and newer version of *Force_Ctrl*. These redundant requests are handled by MiRPA in the way outlined in Sec. 3.2. Here, the *Execution* module considers only the first answer and redundant answers are rejected. Now the older module version can be stopped without risking that requests are not served in time. System reconfiguration, system reboot, and thus expensive down times are avoided. The *Execution* module, which acts as a client in this case, does not notice the module update.

The authors' web pages [56] provides a number of video clips showing demanding robot applications, which have been realized with MiRPA-based robot control systems.

6 Conclusion

MiRPA features a *thin* real-time middleware, which supports client/server as well as publisher/subscriber communication, i.e., worst-case latencies for intercommunication procedures among several software modules can be guaranteed. By means of MiRPA's real-time name service, it is possible to exchange modules during runtime. In combination with its router functionality, it is very easy to set-up real-time capable control control systems consisting of several computing nodes. As MiRPA is able to administer redundant software modules, it is possible to integrate redundancies in a very systematic way. Of course, this is not restricted to redundant servers — also redundant communication paths can be used.

Compared to other existing solutions, MiRPA does not offer many services — it is only a very *thin* high-performance software layer between the operating system and the applications, and it is only responsible for *all* inter-module communication processes, such that all application software modules only have one communication partner: the middleware. Moreover, reliable intercommunication processes are ensured by MiRPA's active deadlock avoidance. As deadlock hazards are detected and avoided automatically, deadlocks cannot corrupt any overall system based on MiRPA. In addition, all communication steps are observed by a watchdog, such that corrupted and/or overloaded software modules can be recognized.

Extensions of the concepts and results presented here are shown in [42]. Furthermore, applications and analyzes are presented in [43, 45, 44]. The future work of the authors will be to extend the set of supported network protocols and operating

systems for MiRPA, such that it can also be used in more heterogenous hard- and software environments as it often is desirable in practice.

Acknowledgements. The authors are deeply indebted to the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), whose funding of the Sonderforschungsbereich 562 (SFB, Collaborative Research Center 562) made this work possible. All the fruitful discussions with Yannick Dadji, Karsten Diethers, Peter Nnamdi Kohn, Jochen Maaß, Markus Olschewski, and Jens Steiner are highly appreciated. Furthermore, the authors would like to thank *QNX Software Systems* for providing free software licenses.

References

1. The Player Project. Free software tools for robot and sensor applications (2009), `http://playerstage.sourceforge.net` (accessed: July 22, 2009)
2. The orca-robotics project. Orca: Components for robotics (2009), `http://orca-robotics.sourceforge.net` (accessed: July 22, 2009)
3. MARIE. Mobile and autonomous robotics integration (2009), `http://marie.sourceforge.net` (accessed: July 22, 2009)
4. Emmerich, W.: Software engineering and middleware: A roadmap. In: Fikelstein, A. (ed.) The Future of Software Engeneering, pp. 117–129. ACM Press, Limerick (2000)
5. Myerson, J.M.: The Complete Book of Middleware, 1st edn. Auerbach Publications, Boca Raton (2002)
6. Geihs, K.: Middleware challenges ahead. IEEE Computer 34(6), 24–31 (2001)
7. Schmidt, D.C.: Middleware for real-time and embedded systems. Communications of the ACM 45(6), 43–48 (2002)
8. Object Management Group, Inc., 140 Kendrick Street, Building A, Suite 300, Needham, MA 02494, USA (2008), `http://www.omg.org` (accessed: December 15, 2008)
9. Fay-Wolfe, V., DiPippo, L.C., Cooper, G., Johnston, R., Kortmann, P., Thuraisingham, B.: Real-time corba. IEEE Transactions on Parallel and Distributed Systems 11(10), 1073–1089 (2000)
10. Pyrali, I., Schmidt, D.C., Cytron, R.K.: Techniques for enhancing real-time corba quality of service. Proc. of the IEEE 91(7), 1070–1085 (2003)
11. Schmidt, D.C., Levine, D.L., Mungee, S.: The design of the TAO real-time object request broker. Computer Communications 21(4), 294–324 (1998)
12. Schmidt, D.C., Kuhns, F.: An overview of the real-time CORBA specification. Computer 33(6), 56–63 (2000)
13. Schmidt, D.C., Deshpande, M., O'Ryan, C.: Operating system performance in support of real-time middleware. In: Proc. of the seventh IEEE International Workshop on Object-Oriented Real-Time Dependable Systems, San Diego, CA, USA, pp. 199–206 (2002)
14. Objective Interface Systems, Inc., 220 Spring Street, Suite 530, Herndon, VA 20170-6201, USA (2009), `http://www.ois.com` (accessed: August 9, 2009)
15. OpenRTM-aist. RT Middleware (2009), `http://www.is.aist.go.jp/rt/OpenRTM-aist` (accessed: August 9, 2009)
16. Callison, R., Butler, D.: Real-time corba trade study. Technical Report D204-31159, The Boeing Company (1999)
17. Tuma, P., Buble, A.: Overview of the corba performance. In: Proc. of the 2002 EurOpen (2002)

18. Polze, A., Sha, L.: Real-time programming with CORBA. In: Proc. of the 24th Euromicro Conference, Network Computing Workshop, Vasteras, Sweden, vol. 2, pp. 997–1004 (1998)
19. Polze, A., Schwarz, J., Wehner, K., Sha, L.: Integration of CORBA services with a dynamic real-time architecture. In: Proc. of the sixth IEEE Real-Time Technology and Applications Symposium, Washington, D.C., USA, pp. 198–207 (2000)
20. Sha, L., Rajkumar, R., Gagliardi, M.: The simplex architecture: An approach to build evolving industrial computing systems. In: Proc. of the ISSAT Conference on Reliability (1994)
21. Rajkumar, R., Gagliardi, M., Sha, L.: The real-time publisher/subscriber inter-process communication model for distributed real-time systems: Design and implementation. In: Proc. of the IEEE Real-Time Technology and Applications Symposium, Chicago, IL, USA, pp. 66–75 (1995)
22. Seto, D., Krogh, B.H., Sha, L., Chutinan, A.: Dynamic control system upgrade using the simplex architecture. IEEE Control Systems Magazine 18(4), 72–80 (1998)
23. Montgomery, J.: A model for updating real-time applications. Real-Time Systems 27(2), 169–189 (2004)
24. Real-Time Innovations, 385 Moffett Park Drive, Sunnyvale, CA 94089, USA. Homepage (2008), `http://www.rti.com` (accessed: December 15, 2008)
25. Yau, S.S., Karim, F.: An adaptive middleware for context-sensitive communications for real-time applications in ubiquitous compuing environments. Real-Time Systems 26(1), 29–61 (2004)
26. Kon, F., Román, M., Liu, P., Mao, J., Yamane, T., Magalhaes, L.C., Campbell, R.H.: Monitoring, security, and dynamic configuration with the *dynamicTAO* reflective ORB. In: Coulson, G., Sventek, J. (eds.) Middleware 2000. LNCS, vol. 1795, pp. 121–143. Springer, Heidelberg (2000)
27. Hauck, F.J., Meier, E., Becker, U., Geier, M., Rastofer, U., Steckermeier, M.: A middleware architecture for scalable, qos-aware and self-organizing global services. In: Linnhoff-Popien, C., Hegering, H.-G. (eds.) Proc. of the third IFIP/GI Conf. on Trends Towards a Universal Services Market, Trends in Distributed Systems: Towards a Universal Service Market, pp. 117–129. Springer, Munich (2000)
28. Murphy, A.L., Picco, G.P., Roman, G.-C.: LIME: A middleware for physical and logical mobility. In: Proc. of the 21st International Conference on Distributed Computing Systems, Mesa, AZ, USA, pp. 524–533 (2004)
29. Handorean, R., Payton, J., Julien, C., Roman, G.-C.: Coordination middleware supporting rapid deployment of ad hoc mobile systems. In: Proc. of the 23rd International Conference on Distributed Computing Systems Workshop, Providence, RI, USA, pp. 362–368 (2003)
30. Hermann, K.: MESHMdl — A middleware for self-organization in ad hoc networks. In: Proc. of the first International Workshop on Mobile Distributed Computing, Providence, RI, USA, p. 446 (2003)
31. Bäuml, B., Hirzinger, G.: Agile robot development (aRD): A pragmatic approach to robotic software. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, pp. 3741–3748 (2006)
32. OROCOS Homepage. Open robot control software (2002), `http://www.orocos.org` (accessed: December 15, 2008)
33. Bruyninckx, H., Soetens, P., Koninckx, B.: The real-time motion core of the orocos project. In: Proc. of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, pp. 2766–2771 (2003)

34. Brooks, A., Kaupp, T., Makarenko, A., Williams, S., Orebäck, A.: Towards component-based robotics. In: Proc. of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, pp. 163–168 (2005)
35. Vaughan, R.T., Gerkey, B.P., Howard, A.: On device abstractions for portable, reusable robot code. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, vol. 3, pp. 2421–2427 (2003)
36. Microsoft Corporation, 1 Microsoft Way, Redmond, WA 98052-7329, USA. Microsoft robotics studio (2008), http://www.microsoft.com/robotics (accessed: December 15, 2008)
37. California Institute of Technology/Jet Propulsion Laboratory, M/S 198-219, 4800 Oak Grove Drive, Pasadena, CA 91109, USA (2009), http://claraty.jpl.nasa.gov (accessed: August 12, 2009)
38. Utz, H., Sablatnög, S., Enderle, S., Kraetzschmar, G.: Miro — Middleware for mobile robot applications. IEEE Trans. on Robotics and Automation 18(4), 493–497 (2002)
39. Finkemeyer, B., Borchard, M., Wahl, F.M.: A robot control architecture based on an object server. In: IASTED International Conference Robotics and Manufacturing, Cancun, Mexico, pp. 36–40 (2001)
40. Finkemeyer, B., Kröger, T., Wahl, F.M.: Executing assembly tasks specified by manipulation primitive nets. Advanced Robotics 19(5), 591–611 (2005)
41. Diethers, K., Finkemeyer, B., Kohn, N.: Realizing open control software for high dynamic processes with a middleware. It - Information Technology 46(1), 39–47 (2004)
42. Dadji, Y., Michalik, H., Kohn, P.N., Steiner, J., Beckmann, G., Möglich, T., Varchmin, J.-U.: A Communication Architecture for Distributed Real-Time Robot Control. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 213–231. Springer, Heidelberg (2010)
43. Maaß, J., Dietrich, F., Hesselbach, J.: RCA562: Control Architecture for Parallel Kinematic Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 315–331. Springer, Heidelberg (2010)
44. Steiner, J., Diethers, K., Goltz, U.: Model Based Quality Assurance for a Robotic Software Architecture. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 373–389. Springer, Heidelberg (2010)
45. Steiner, J., Maaß, J., Goltz, U.: Self-Management within a Software Architecture for Parallel Kinematic Machines. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 355–371. Springer, Heidelberg (2010)
46. RTnet — Hard Real-Time Networking for Real-Time Linux (2008), http://www.rtnet.org (accessed: December 15, 2008)
47. Xenomai (2008), http://www.xenomai.org (accessed: December 15, 2008)
48. QNX Software Systems, 175 Terence Matthews Crescent, Ottawa, Ontario, Canada, K2M 1W8 (2009), http://www.qnx.com (accessed: October 23, 2009)
49. Wind River Systems, Inc., 500 Wind River Way, Alameda, CA 94501, USA (2009), http://www.windriver.com (accessed: August 12, 2009)
50. Kröger, T., Finkemeyer, B., Wahl, F.M.: Manipulation Primitives — A Universal Interface Between Sensor-Based Motion Control and Robot Programming. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 293–313. Springer, Heidelberg (2010)
51. Finkemeyer, B., Kröger, T., Kubus, D., Olschewski, M., Wahl, F.M.: MiRPA: Middleware for robotic and process control applications. In: Workshop on Measures and Procedures for the Evaluation of Robot Architectures and Middleware at the IEEE/RSJ International Conference on Intellegent Robots and Systems, San Diego, CA, USA, pp. 78–93 (2007)

52. Kröger, T., Finkemeyer, B., Winkelbach, S., Molkenstruck, S., Eble, L.-O., Wahl, F.M.: A manipulator plays Jenga. IEEE Robotics and Automation Magazine 15(3), 79–84 (2008)
53. Stäubli Faverges SCA, Place Robert Stäubli BP 70, 74210 Faverges (Annecy), France (2008) http://www.staubli.com/en/robotics (accessed: December 15, 2008)
54. Osypiuk, R., Kröger, T.: Parallel Stiffness Actuators with Six Degrees of Freedom for Efficient Force/Torque Control Applications. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 275–291. Springer, Heidelberg (2010)
55. The MathWorks Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA (2008), http://www.mathworks.com (accessed: December 15, 2008)
56. Institut für Robotik und Prozessinformatik at the Technische Universität Carolo-Wilhelmina zu Braunschweig, Mühlenpfordtstr. 23, D-38106 Braunschweig, Germany (2008), http://www.rob.tu-bs.de/en (accessed: December 15, 2008)

A Communication Architecture for Distributed Real-Time Robot Control

Yannick Dadji, Harald Michalik, Nnamdi Kohn, Jens Steiner, Guido Beckmann, Tobias Möglich, and Jörn-Uwe Varchmin

Abstract. Due to their continuous development in terms of performance and also reliability, standard PCs offer an adequate platform as basis for the development and implementation of control systems. The efficient and robust control of highly dynamic systems requires the implementation of very short control cycles ($f_{cycle} \geq$ 1 kHz). This requires real-time and high performance communication mechanisms implemented in software with dedicated hardware support providing response times in the range of a couple of μs. Such time constraints are typically not covered by commercial off-the-shelf (COTS) products in the full scale. Therefore, we introduce a communication architecture for a PC-based control platform that spans the complete chain from software development to integration of sensor and actuator devices with respect to real-time and performance requirements for high end control

Yannick Dadji · Harald Michalik · Tobias Möglich
Technische Universität Braunschweig, Institute of Computer and Communication Network Engineering, Hans-Sommer-Straße 66, 38106 Braunschweig, Germany
e-mail: {dadji,michalik,moeglich}@ida.ing.tu-bs.de

Nnamdi Kohn
IAV GmbH, Rockwellstraße 16, 38518 Gifhorn, Germany
e-mail: nnamdi.kohn@iav.de

Jens Steiner
Technische Universität Braunschweig, Institute for Programming and Reactive Systems, Mühlenpfordtstraße 23, 38106 Braunschweig, Germany
e-mail: steiner@ips.cs.tu-bs.de

Guido Beckmann
Beckhoff Automation GmbH, Eiserstraße 5, 33415 Verl, Germany
e-mail: g.beckmann@beckhoff.de

Jörn-Uwe Varchmin
Technische Universität Braunschweig, Institute of Electrical Measurement and Fundamental Electrical Engineering, Hans-Sommer-Straße 66, 38106 Braunschweig, Germany
e-mail: j-u.varchmin@tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 213–231.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

applications. Its key components are the distributed middleware MiRPA-XD and the Industrial Automation Protocol (IAP) applied on top of the IEEE1394 standard. This approach allows the achievement of flexible and high performance (up to 8 kHz) control systems at low cost using standard PC technology.

1 Introduction

In the last decade, PC-based systems have been more and more applied in the area of robot control [1] [2] [3]. Besides the advantage in terms of low cost due to the use of standard components and interfaces, standard PCs offer a continuously upgrading hardware platform thus enabling the design of robot control systems always based on state-of-the-art hardware technology. However, implementing a PC-based control solution requires special attention to the function and capabilities of the operating system. To be a viable alternative to PLCs (programmable logic controller) in a control application, PC-based control system must provide deterministic operations. The control function must be treated with highest system priority in order to ensure repeatable responses and low jitter. Furthermore, the available computing power is to be used for the control functions, utilizing just a minimum for OS features. The real-time microkernel based operating system QNX [4], developed by QSSL, meets all of these requirements.

Using PC technology opens the door to the the development of more flexible robot control approaches. Introduced in 1981 by Mason [5] and taken up by De Schutter [6], the task frame formalism is a major step towards task orientated assembly programming [7]. Unlike conventional motion description, the motion is defined within a task frame, which is allowed to be placed freely everywhere in the robots world. The skill primitive programming interface allows the robot operator to define a separate control algorithm for each of the six DOF the task frame provides, causing hybrid moves on the robot's end effector. To get advantage of the skill primitive programming in the robot control, the development of a suitable control architecture is necessary. The RCA562 (Robot Control Architecture 562) [8, 9, 10], developed within the scope of the collaborative research center 562, is such an architecture. Its design allows motion programming in the task frame formalism and integration of arbitrary sensors to provide hybrid control schemes. The strengths of this approach are its modularity and its ability to modify the hybrid control scheme on runtime according to the current skill primitive.

To handle the complexity of the software development, a communication middleware is applied. The middleware must support the modular software structure of the RCA562. It handles all details related to the communication between software modules and thus simplifies the implementation, the potential expansions and the maintenance of the control system. The control of parallel kinematics is subject to additional requirements with respect to performance. Parallel kinematics can achieve movements with very high velocities and accelerations (e.g TRIGLIDE: up to 10 g). To ensure an accurate motion control and short reaction time to unpredictable events, short control cycles (typically ($f_{cycle} \geq 1$ kHz)) are required. Therefore,

the communication infrastructure must provide high performance and deterministic mechanisms both for the Inter Process Communication (IPC) on the control PC and the communication between control PC and actuators/sensors. The communication latency must be kept as small as possible, such that the overall system performance is not affected. Depending on the control task, synchronisation mechanisms and response times in the order of a couple of microseconds are required.

Numerous frameworks for the development of PC based control software are available in the literature. The aRD (agile Robot Development) framework [11] provides a distributed software concept especially designed to support the development of complex robotic systems. It is based on the abstract view of a robotic system as being a decentral net of calculation blocks and communication links. Control frequencies up to 1 kHz can be achieved, but software development flexibility, as provided by middleware based systems, is not available. The user needs to care for details, e.g. for the communication ports defined in the system. Other Framework representatives are CORBA [12], OROCOS [13], OSACA [14], MARIE [15], ORCA [16], MIRO [17], and Player [18]. These frameworks were successfully used in applications, where realtime constraints are relatively soft, e.g. in the field of mobile robotics. The current implementations of these frameworks have not proven that they can reach realtime rates above some 100 Hz in complex applications with high numbers of DOFs [11].

Therefore, we introduce a suitable framework to cover the requirements mentioned above. This framework consists of the real-time middleware MIRPA-XD (Middleware for Robotic and Process Control Applications- eXtended and Distributed) [19, 20, 3, 21] and the dedicated protocol IAP (Industrial Automation Protocol) [22, 23] developed to assure a deterministic data exchange between the control PC and external actuators/sensors. Both have been implemented in ANSI C with QNX Neutrino as target real-time operating system.

The following section will introduce an overview of the communication framework. The IAP protocol will be outlined in the third section. The features of the communication middleware MiRPA-XD will be addessed in section four.

2 Overview of the Communication Framework

The structure of the communication framework is outlined in Fig. 1. To meet the real-time requirements of the control software, the microkernel based OS QNX Neutrino is used as operating system on the control PC. In order to support the development of a modular control software, the middldeware MIRPA-XD is applied as key feature. MIRPA-XD handles all details related to the communication between the software modules. MiRPA-XD has been developed using the conceptual idea of the existing middleware MiRPA [24] for the realization of synchronous and asynchronous message passing. But MiRPA-XD represents a complete new design with new features and a lean implementation optimized for the RCS562 applications. The new design was implemented in two steps: 1. MiRPA-X for centralized control and 2. MiRPA-XD for distributed control support. Its lean design allows the

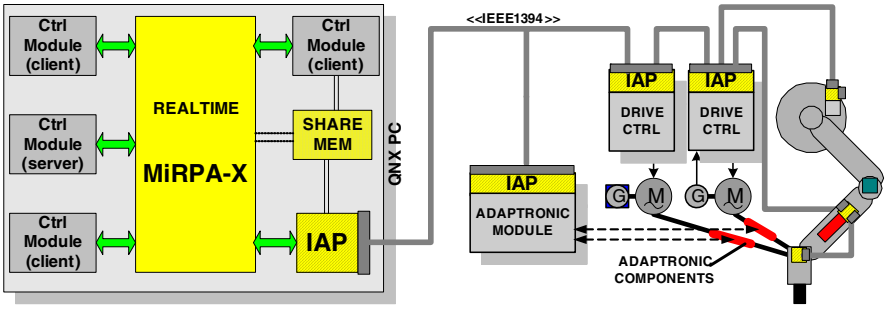


Fig. 1. Software and communication architecture of the parallel robot control system.

implementation of low latency synchronisation and message passing based communication mechanisms. In the MIRPA-XD environment the control modules are implemented as client or server processes.

To guarantee a fix control cycle time and deterministic data exchange between the control software and external actuators/sensors, the dedicated IAP protocol is applied. The IAP has been designed to benefit from the underlying IEEE1394 standard [25] with respect to real-time data transfer and high data throughput. It is capable of efficient operations with very short cycle times (down to 125 μs, defined by the IEEE1394 Standard) and to ensures hard real-time, bandwidth and synchronism constraints. The IAP is implemented on both the control PC (IAP-Master) and the external robot communication nodes (IAP-Slaves), which act as link to the drive controllers. The IAP defines a communication cycle, in which the Master sends the set values and receives the actual sensor values from the IAP-Slaves. A detailed description of the IAP protocol follows in the next section.

2.1 Selection of the Bus System

Using the IEEE1394 standard as link between the control PC and the drive controller was the result of intensive evaluation of high speed communication systems [22]. The following systems SERCOS [26], MACRO [27], USB [28], Fast and Gigabit Ethernet [29], IEEE1394 standard [25] and Fibre Channel were examined for their suitability. The evaluation criteria were transmission bandwidth, achievable cycle time, synchronization mechanism, parameter and process data channels, ability for data cross-connection traffic and distribution of the system. The evaluation of these systems resulted in clear advantages of the IEEE1394 standard over all the other communication systems [22].

Eight years after starting the evaluation of the system architecture, the requirements to the communication system are still valid. High performance with short cylce times and high precision synchronization are the main features of modern communication systems. Nevertheless the field bus world has changed. In the

meantime Ethernet is established as a successor of the standard field bus systems. To achieve this, extensions of the basic Ethernet standard IEEE 802.3 according to determinism, synchronization and efficiency. The IEC 61158 [30] series specifies several solutions for Industrial Ethernet networks. There are many different approaches that try and provide real-time capability for Ethernet: for example the CSMA/CD media access mechanism is disabled via higher level protocol layers and replaced by the time slice mechanism or polling; other propositions use special switches that distribute Ethernet packets in a precisely controlled timely manner. Many Industrial Ethernet protocols have been proposed by different vendors. The most relevant systems fulfilling the required claims are EtherCAT [31], SERCOS III [26] and PROFINET IRT [31].

3 The Industrial Automation Protocol (IAP)

The Industrial Automation Protocol was developed to serve as a communication protocol for IEEE1394 serial buses [22, 23, 32]. In our system it is responsible for asynchronous and isochronous data transmissions between the main control PC and sensor/actuator devices in the field. The IAP realizes cyclic communication up to 8 kHz and is responsible for the adherence to real-time and synchronisation constraints [33]. In addition it provides the following functionalities:

- Definition of participant classes (Master, Node, Control)
- Management of resources (bandwidth and channels)
- Network management (monitoring and causing state changes)
- Global parameter access (read and write distributed parameter sets)
- System wide synchronization (via cycle start telegram)
- Self-Configuration (automatic configuration of all participants)
- Monitoring (handling missing or late telegrams)

There are different classes of IAP participants with different behaviour. Distributed processing elements, which control sensor or actuator devices, run an *IAP Node*, which must be initialized after start-up with the current system configuration issued by the *IAP Master*. This *IAP Master* manages available bandwidth and communication channels. These are assigned to the *IAP Nodes* during the system configuration phase. During active operation the *IAP Master* maintains synchronism in the distributed system by trigerring a *Cycle Start Telegram* (CST) every 125 μs. The *IAP Master* communicates with the control application via shared memory areas and by means of the MiRPA-XD token mechanism explained in Sect. 4.1. It sends nominal values to actuators via a *Master Data Telegram* (MDT) and extracts the actual values sent by *IAP Nodes* in *Device Data Telegrams* (DDT). Fig. 2 shows a 1 kHz communication cycle with asynchronous MDT and DDTs. The controlled robot in that case is the TRIGLIDE demonstrator with 3 linear drives, 1 analogous, and 1 digital I/O node. The rightmost CST starts a new cycle where the next MDT would follow.

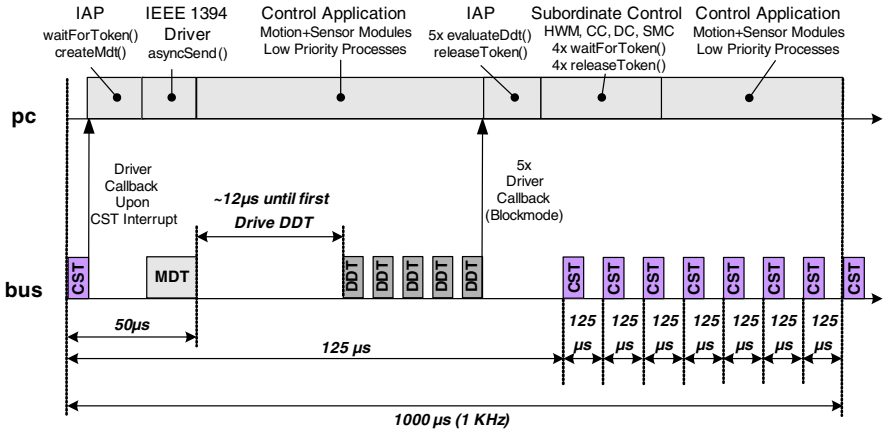


Fig. 2. Communication cycle (1 kHz).

3.1 Synchronisation

Asynchronous telegrams could in principle cause a delay of the next following CST. In our system only the first CST of a communication cycle is used for synchronisation (the one before the MDT). For a 1 kHz communication cycle this is every 8th CST (see Fig. 2). It has a jitter smaller than 10 nanoseconds [25] because nothing is sent in the 125 μs before it. Thus, this CST provides a very precise basis for synchronization.

IAP Nodes use the MDT to synchronize their local applications and to ensure nominal values are made valid and actual values are read at a system wide fixed time. MDT creation and transmission is issued in reaction to a cycle start interrupt that is handled with the highest priority in the system. Due to this, the jitter of the MDT is also very small ($\leq 3\ \mu s$). By using the MDT instead of the CST as a synchronization signal, the *IAP Nodes* save execution time by masking the CST interrupt.

Fig. 2 also shows that the highest priority processes in the system use the token mechanism of MiRPA-XD to ensure a deterministic and clock synchronized execution of essential functions (IAP, HWM - *Hardware Monitoring*, CC - *Control Core*, DC - *Drive Control*, SMC - *Smart Machine Control*). When none of these high priority processes possesses the token, all other calculations may take place (*Motion/Sensor Modules*). More details about the control system and its use of the communication system described here can be found in [8].

3.2 Self-configuration

The IAP operates as a protocol layer in between the IEEE1394 driver and the robot control software. The control engineer does not need any knowledge about the driver API or complex FireWire specifics to configure the communication with the

sensors and actuators. Instead, just some parameters have to be specified within an XML file:

- Communication mode (asynchronous or isochronous)
- Cycle frequency (1-8000 Hz)
- Bandwidth limitation
- For each node
  - Extended Unique Identifier (EUI)
  - Specification of incoming data (nominal values)
  - Specification of outgoing data (actual values)

According to those values the *IAP Master* configures the FireWire driver, reserves communication channels and bandwidth, and by means of MIRPA-XD mechanisms creates shared memory areas for the communication with the robot control software modules. It also configures the *IAP Nodes* by writing configuration data directly into their remote parameter sets.

3.3 Monitoring

During active operation the cyclic communication as depicted in Fig. 2 takes place. The *IAP Master* recognizes, if an *IAP Node* stops working properly and consequently sends delayed DDTs or no DDTs at all. The control application will be notified accordingly and tries to compensate a certain amount of missing DDTs while initiating a braking strategy. If a certain error threshhold is surpassed, the *IAP Master* will enter an error state and will stop all nodes using a network management telegram.

4 Middleware MiRPA-XD

MiRPA-XD [21], in the following referred to as XD, is the distributed version of the MiRPA-X [32] middldeware. XD was developed to cover the structural and real-time requirements (as mentioned above in section 1) as well as the growing demand for scalabe computing power as requested by complex control algorithms, such as those implemented within the SFB562 [8, 9, 10]. XD allows the use of multiple PCs to develop both real-time and distributed robot control software. The distribution of control software on available PC nodes is hereby transparent for the user. Using XD services, the user client and server applications must specify neither details about the communication partner nor the PC node on which they should be executed, since this information is managed by XD (Fig. 3a).

Fig. 3b) depicts the architecture of XD. It consists of an object server (OBS), a remote control unit (RC), a token controlller (TC) and a system monitor (SM). The OBS is the core functionality of XD. It starts and initializes the TC and the SM if required by the user application. During runtime, the OBS provides real-time communication between client and server application modules via message

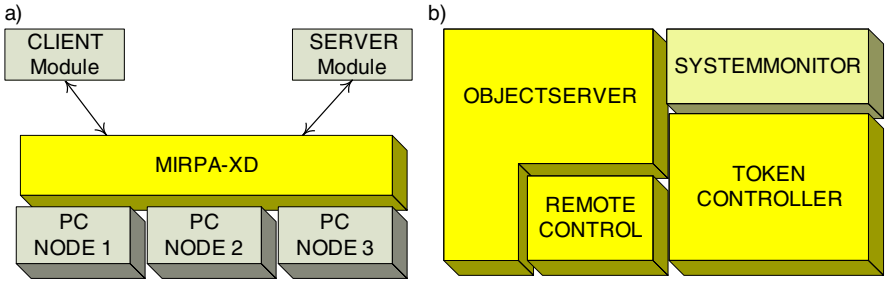


Fig. 3. a)Transparent resource utilization in the distributed environment, b) structure of the real time middleware MiRPA-XD.

passing. The RC unit is responsible for extending the message passing communication over the physical PC boundary. Furthermore, it provides the hot-plug capability of distributed XD entities over the IEEE1394 network. The TC provides an adjustable and reliable communication bandwidth for user applications (called "token-threads" in this context) in a control cycle. It operates at the highest available user priority level. By means of a successive scheduling of all token-threads within a control cycle, the TC provides secure data exchange between software modules using shared memory mechanisms. The SM delivers an image of ongoing system activities at runtime. It collects topology information and execution times of controlled tasks and stores these in a database, which is used as platform for the integration of self-x properties [10, 34] in the robot control system.

Communication in the XD environment is based on message passing and shared memory, whereas the latter may only be used locally. Each user process may be registered as client and/or server application. One of the key features of XD is the ability to hide the identity of servers from clients; so every data transfer is content-based, not structure-based. This allows even dynamic modifications of the whole control system structure without the necessity of recompiling or restarting the system components involved.

4.1 Communication and Synchronisation Mechanisms

The idea of realizing synchronous and asynchrounous communication mechanisms via synchronous message passing in XD was derived from the middleware MiRPA [35, 24, 36]. In the MiRPA approach an object server is basically used as a core communication entity which functions as the link between user processes by handling all message passing based communication. However, motivated by the need to reduce the middleware communication overhead and improve the performance of its communication mechanisms, the object server of XD handles the messages differently. Concerning message passing handling, XD distinguishes a configuration and a runtime phase. In the configuration phase a message must first be registered before it may be handled by XD in the runtime phase. During the registration, XD

performs the internal message indexing, hashing calculation and creates a message object, which contains both protocol data and required memory space for the user data. Then it saves the hash value in the message object and returns a message handle and a pointer object for the dedicated message data memory to the user process. In the runtime phase, the message handle is used to refer to the registered message. Furthermore, the user directly accesses the message data memory. This way XD avoids additional memory to memory copying of user data while processing the sending and receiving of messages. This is one means by which high performance communication is achieved. For data payload up to 2 kByte the message trasmission time in a local system is $\leq 4\ \mu s$ for asynchronous communication and $\leq 6\ \mu s$ for synchronous communication.

Fig. 4 shows the general message types which can be communicated between a server module, a client module and the XD object server. MiRPA-X supports four different asynchronous message types:

- REQUEST / ANSWER - the requesting client blocks until an answer is received. This type is used for the synchronous client/server communication
- COMMAND - the issuing client continues operations. This type is used for asynchronous the publisher/subscriber communication model.
- CONFIG - registration of messages and shared memory access.
- CONTROL - retrieval of statistical information about the control system system directly from XD.

The functioning of both COMMAND and REQUEST mechanisms basically follows the design described in [35, 24] and specific variations are presented in [19, 32].

In addition to message passing, XD supports token controlled communication. This mechanism allows token-threads to be automatically scheduled for processing in a sequential and cyclic manner at the highest user priority. This is particularly advantageous for control applications operating on data in a cyclical manner using several processing steps, each step expecting the result of the previous one. Through

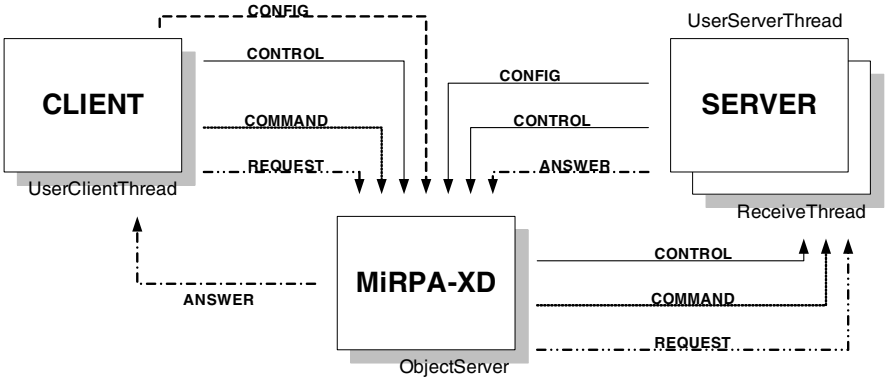


Fig. 4. MiRPA-XD message passing.

the utilization of sequential scheduling, the token controlled communication additionally allows a synchronized and high performance data exchange between the token-threads on the basis of shared memory. After the activation of the token controlled communication all token-threads wait in a blocked state to be activated by the token controller. The TC performs the scheduling by sending the token to the token-threads in a sequential order. Upon reception of the token, token-threads unblock and process user operations. Upon completion of processing token-threads send the token back to the TC and blocks again, waiting for the next activation. The TC then sends the token to the next token-thread in the sequence and so on.

XD further offers various mechanisms for synchronisation of software modules on the local system. It supports the use of condition variables, mutual exclusion mechanisms (mutex) and signals. A detailed description of XD synchonisation mechanisms and token controller is presented in [32].

4.2 Remote Control, RC

To cover the growing computing demand generated by complex and time expensive data processing control strategies (i.e. singularity prediction [37, 38], incorporation of vision data [39]), the available computing power must be extended. To reach this goal, two approaches could be suitable: a multicore platform or a network of PCs. The multicore platform has the advantage to provide enlarged computing power through concurrency; it offers high system density, but is not extendable. On the other side, a network of PCs has a lower performance due to the network communication overhead. But it offers more system resources (i.e. memory, IO interface, ...) and allows clear system design, where special heavy load control funtionalities (camera data processing) can be assigned to dedicated PCs. Furthermore, the computing power in a network of PCs is scalable. XD supports application development on both the multicore platform and the network of PCs. While the task management on the multicore platform is handled by the operating system QNX, XD manages the task distribution on the network of PC by means of its remote control unit.

4.2.1 Architecture of the Distributed System

The design of the distributed architecture has mainly been influenced by the required real-time capability of the distributed control components. The real-time behavior of a remote application module implies that all response time constraints are fulfilled, regardless of the present amount of communication and computation. For control systems the time constraints are usually in the range of a few ten μs.

Originally, the network distribution of XD was designed to be applied over an extension of the IAP protocol. The corresponding master/slave architecture is outlined in Fig. 5 a). In this approach all system participants (control master PC, Slave PC, sensor/actuator nodes) are connected to the same bus. In addition to the real-time data amount to be distributed between the control master PC, sensor and actuator nodes, the IAP would have to rule the communication between the master and the slave PCs. Theoretically this is possible, since there is enough bandwidth

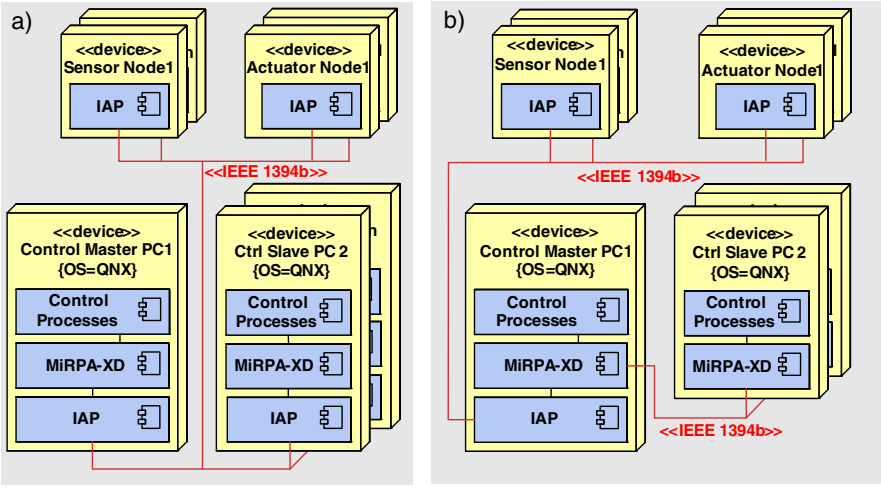


Fig. 5. Network structure of the XD distribution. a) Distribution over the IAP protocol or one Link approach, b) Distribution over a separate communication link or 2 links approach.

available on the bus. But practically this leads to the integration of an event based asynchronous communication in a time triggered communication protocol. In this case conflicts in the protocol layer may happen, for example when a data transmission, issued by a PC slave, delays the transmission of time critical sensor data. To solve this problem, a management component can be integrated in the IAP protocol, which assures the compliance with all real-time requirements of the system while assigning prioritized communication channels to the time critical sensor/actuator data. However, this will be at the cost of potential additional delays within the master/slave PC communication. A better solution is to physically separate the master/slave PC communication from the master PC to sensor/actuator communication. The resulting structure with two separated bus connections to the master PC is outlined in Fig. 5 b). Here the master and the slave PCs are logically linked via XD components. The central components of the control software run on the master PC. Computationally intensive algorithms are encapsulated within server applications and executed on slave PCs. XD allows transparent access of remote resources via message passing.

4.2.2 IEEE1394: A Medium for Distributed Networking

A particularity of our distributed architecture is the physical link connecting master and slaves. In fact, we use the IEEE1394 standard (FireWire) instead of the common Ethernet bus. This design decision was motivated by the need to provide real-time capability in the distributed environment. Indeed, the use of the IEEE1394 standard is favorable for distributed real-time control because of its inherently deterministic network behavior [40]. A data rate of 800 Mbit/s and the efficient bus arbitration

scheme BOSS (Bus Owner Supervisor Selection) allow the implementation of remote procedure calls with low latency.

4.2.3 Remote Resource Registration and Access

Since XD assumes a master/slave communication scheme, the master must always know which resources are available on the distributed system. For this reason, all resource registrations to XD on remote slaves are automatically published to the master PC via a configuration message. It contains the information (resource name, resource ID on the slave PC, slave ID) that is necessary to identify the resource in the distributed environment.

XD perfroms a remote resource access transparent to the user. To issue a request (Fig. 6), a client module must neither specify the server that serves the request message, nor the PC node on which the server runs. XD searches the location of the target server in its internal data base. If the target server resides on the local PC, the message request is immediately delivered. In the other case XD delivers it to the corresponding remote target PC. On the remote PC, the local XD entity then delivers the request to the target server. After the server completes the processing of the request message, it sends back the reply message to the requesting client the same way. Besides the synchronous request, XD also supports asynchronous communication in the distributed environment. To improve the quality of service (QoS) in the distributed system, the prioritized transmission and the automatic retransmission feature (in case of network failure) of the IEEE1394 standard is applied. In order to assure the scalability of the network, XD supports hot-plug of additional computing slaves [21].

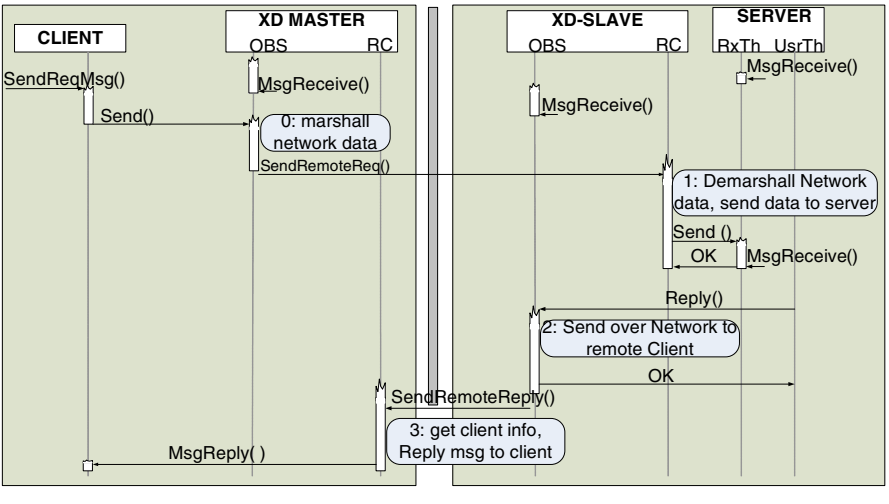


Fig. 6. Access of remote resources via synchronous request feature of XD.

4.3 System Monitor, SM

During the execution of complex motion tasks, the algorithm load usually varies over the time. In the RCA562 a self manager is applied to generate an optimal task distribution pattern and map the control tasks to the different available CPU resources. The article [34] in this book gives detailed information about the integration of this self manager into the RCA562. Previous related publications can be found in [9, 10]. The self manager requires topology information and task execution time information, which must be delivered on the fly by the communication system. For this purpose a system monitoring (SM) has been designed and integrated into the XD.

4.3.1 Design of the System Monitor

From the user's point of view a task is an unambiguously identifiable job. In the control context a task can embed e.g. a motion planning algorithm. From the view of XD, a task is a user program code, that is embedded in two associative functions of the XD API. Associative functions are such, which describe a complete transaction and define the start and end of the task program code. In the XD environment, the token based communication as well as the COMMAND and REQUEST messages may be used to implement user tasks.

The design of the SM has been influenced by two principal requirements. The monitor must be configurable, i.e. the user must be able to specify, which task should be monitored. Further, the monitor should have no impact on the real-time behavior of the system and not or just minimal affect the system performance. The corresponding design is depicted in Fig. 7. To configure the system monitor

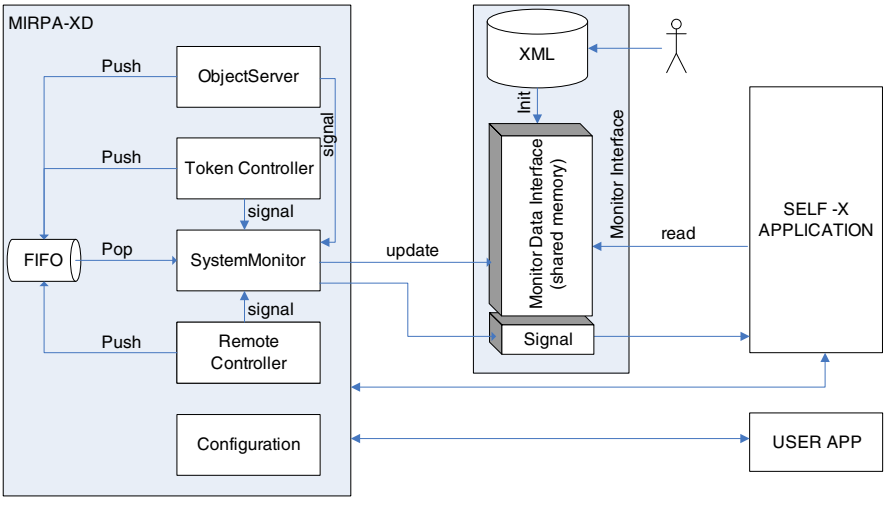


Fig. 7. Architecture of the system monitor.

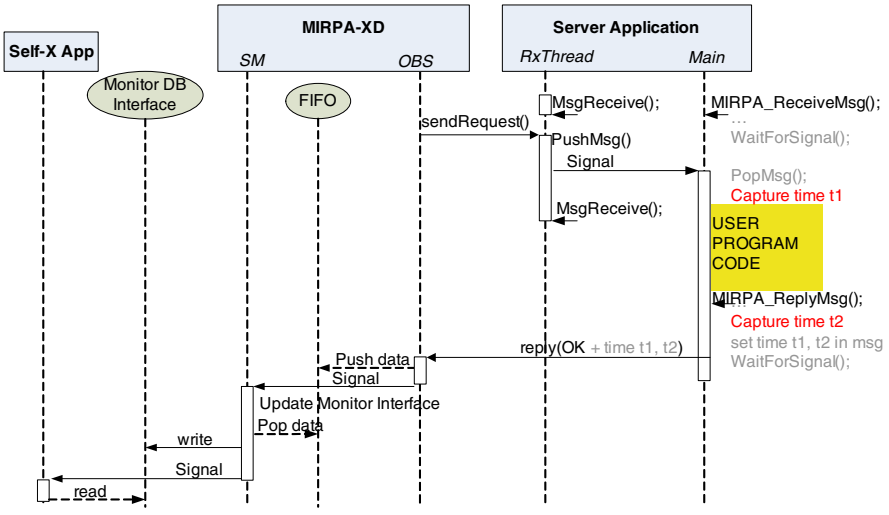


Fig. 8. Task implementation via sysnchonous REQUEST mechanism and capture of task execution time at runtime.

the user specifies tasks to be monitored in a XML configuration file. The monitor uses a shared memory based data interface to exchange monitor data with the self manager.

During user operations, the execution times of registered tasks are collected by the XD components *Object Server*, *Token Controller* and *Remote Controller*. The automatic measurement of task execution time is performed through introduction of time measurement points in the XD program code of the associative functions. In order to not affect XD activities, the collected data is passed to the SM for processing at a lower priority. The data transfer occurs via an entry (push) into the monitor FIFO and a subsequent activation (via a signal) of the SM. After activation, the SM processes the FIFO entries. For each entry it calculates the effective task execution time and saves the results in the monitor data interface. Then it triggers the self manager application, which accesses the monitored data to process the self management procedure.

4.3.2 Task Implementation

The measurement of task execution time is performed through the introduction of time measurement points in the associative functions of the XD API. Fig. 8 illustrates task implementation using XD REQUEST messages. Upon reception of a client request, the server processes the task specified as user program code and replies the results via a REPLY message. In this case the user task is embedded in the API functions MIRPA_ReceiveMsg() and MIRPA_ReplyMsg(). To capture the task execution time, XD just integrates two time measurement points in the code of the API functions. The first point (*CaptureTime(t1)*) will be set in the receiving

function MIRPA_ReceiveMsg() immediately after the reception of the request message. The second point (*CaptureTime(t2)*) will be set at the beginning of the code of the reply function MIRPA_ReplMsg(). Both time values are then compiled in the reply message and sent back to the OBS. Upon reception the OBS makes a corresponding entry in the monitor FIFO and informs the SM via a signal. The SM therefore processes a prior configured self-x related action. This may either be the simple actualization of the monitoring data base or the activation of an alarm in the case of advanced configuration. Since the system monitor is realized as low priority process, it cannot delay the execution of real-time communication.

The task implementation within a token-thread is similar. Here, the TC assures the capture of task execution time, cares for data entry in the monitor FIFO and activates the system monitor for processing. While implementing tasks via COMMAND messages, an automatic measurement of execution time is not possible. Since the server module implementing the task cannot reply a COMMAND message, the measured task execution time cannot be passed to the system monitor without an additional communication effort. Nevertheless, the task processing is generally not affected in this case, since capture and transmission of time information is only performed after completion of the task processing.

4.4 Performance and Application

Since XD is a key component for the software development, the latency caused by its communication mechanisms, especially REQUEST and COMMAND is an important index for the overall system performance. Performance measurements were made on a Pentium 3 machine with 3 GHz clock frequency running the operating system QNX Neutrino. Messages with 100 bytes of data payload have been taken into consideration representing typical cases. The results of the measurement are outlined in Table 1. On the local system we achieve very short latencies for the delivery of COMMAND ($\leq 4\ \mu s$) and REQUEST ($\leq 6\ \mu s$ with reply) messages. On the network we achieve a latency of 50 μs for the delivery of a COMMAND message. The round trip latency for REQUEST message is 94 μs. In both cases the jitter is $\leq 3\ \mu s$. For the same data size a round trip latency of 120 μs with the worst case value of 140 μs was measured in in MiRPA [24].

Table 1. Performance measurement of message based communication mechanisms of MiRPA-XD for 100 bytes data payload (values in μs).

	Local		**Network**	
	Latency	Jitter	Latency	Jitter
COMMAND	3.3	1.6	50	3
REQUEST	5.4	2.2	94	3

The application of the IAP protocol on top of the IEEE1394 standard allows the realization of control cycle frequencies up to 8 kHz. Within the SFB562 however, the used drive controller hardware limits the achievable control frequency to 4 kHz.

The RCA562 control system, referenced in this book in [41], has been design on top of this communication framework. Currently, first control scheme including distributed software component and self management properties are in implementation.

5 Conclusion

The use of PC based systems allows the design of a generic robot control architecture with state-of-the-art hardware platforms. The communication architecture we introduced supports the complete development chain from the software design up to the integration of sensor and actuator devices. The real-time midleware MiRPA-XD is applied as communication middleware. It enables a modular software development and supports synchronous and asynchronous message passing based communication. MiRPA-XD additionally provides a secure high performance data exchange using shared memory mechanisms and token controlled communication. MiRPA-XD enables the update of software modules during runtime. In order to support the integration of self-x properties in the control applications, a system monitoring component is integrated in the MiPRA-XD environment. It automatically captures the topological information, measures the execution time of user tasks on local and remote PCs and delivers the information to the user on demand.

Compared to other middleware systems, MiRPA-XD offers inter-process communication mechanisms with very low latency for local as well as for distributed environments. This is due to the performance oriented software design applied to MiRPA-XD.

To connect sensors and actuators to the system the Industrial Automation Protocol (IAP) was developed. It is built on the top of the IEEE1394 serial bus and manages real-time exchange of control data between the control PC and the sensors/actuators. The IAP supports control frequencies up to 8 kHz.

The platform presented has successfully been applied for the development of new control approaches for parallel kinematics within the scope of the SFB562 at the technical university of Braunschweig. The achievable control frequency is application dependent and is currently limited to max 4kHz, due to the processing latency of the drive controller. First distributed control patterns that use the computing power of multiple PCs are currently in implementation.

Acknowledgements. The authors highly appreciate the support given by the German Research Foundation (DFG) within SFB562 and by QNX Software Systems providing free software licenses.

References

1. Gee, D.: The how's and why's of pc based control. In: Conference Record of. Pulp. and Paper Industry Technical Conference, pp. 67–74 (2001)
2. Costescu, N., Loffler, M., Zergeroglu, E., Dawson, D.: Qrobot - a multitasking pc based robot control system. In: Proceedings of the 1998 IEEE International Conference on Control Applications, vol. 2, pp. 892–896 (1998)
3. Dadji, Y., Maass, J., Michalik, H., Möglich, T., Kohn, N., Varchmin, J.-U.: Networked architecture for distributed pc-based robot control systems. In: proceedings of the International Conference on Automation, Robotics and Control (ARCS), Orlando(FL), USA (2008)
4. QNX Neutrino homepage (2009), `www.qnx.com/`
5. Mason, M.T.: Compliance and force control for computer controlled manipulators. IEEE Transactions on Systems, Man and Cybernetics 11(6), 418–432 (1981)
6. De Schutter, J., Van Brussel, H.: Compliant Robot Motion I. A Formalism for Specifying Compliant Motion Tasks. The International Journal of Robotics Research 7(4), 3–17 (1988)
7. Thomas, U., Maaß, J., Hesselbach, J., Wahl, F.M.: Towards a new concept of robot programming in high speed assembly applications. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2005, pp. 3932–3938 (2005)
8. Maaß, J., Kohn, N., Hesselbach, J.: Open modular robot control architecture for assembly using the task frame formalism. International Journal of Advanced Robotic Systems 3(1), 1–10 (2006)
9. Maaß, J., Hesselbach, J., Steiner, J., Goltz, U.: Self-management in a robot control architecture. In: Proceedings of Second International Workshop on Software Development and Integration in Robotics (SDIR 2007), affiliated with ICRA 2007 (2007)
10. Maaß, J., Steiner, J., Amado, A., Hesselbach, J., Huhn, M., Raatz, A.: Self-management in a control architecture for parallel kinematic robots. In: Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2008, Brooklyn, New York, USA (2008)
11. Bäuml, B., Hirzinger, G.: Agile robot development (ard): A pragmatic approach to robotic software. In: Proceedings of the IEEE/RSJ International Conference on Intellegent Robotic Systems (2006)
12. Orfali, R., Harkey, D., Edwards, J.: Instant CORBA. John Wiley & Sons, Chichester (1998)
13. Orocos homepage. open robot control software (2009), `www.orocos.org`
14. Osaca homepage (2009), `www.osaca.org`
15. Cote, C., Letourneau, D., Michaud, F., Valin, J.-M., Brosseau, Y., Raievsky, C., Lemay, M., Tran, V.: Code reusability tools for programming mobile robots. In: Proceedings. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, pp. 1820–1825 (2004)
16. Brooks, A., Kaupp, T., Makarenko, A., Williams, S., Oreback, A.: Towards component-based robotics. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp. 163–168 (2005)
17. Utz, H., Sablatnog, S., Enderle, S., Kraetzschmar, G.: Miro - middleware for mobile robot applications. IEEE Transactions on Robotics and Automation 18(4), 493–497 (2002)
18. Vaughan, R., Gerkey, B., Howard, A.: On device abstractions for portable, reusable robot code. In: Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), vol. 3, pp. 2421–2427 (2003)

19. Dadji, Y., Michalik, H., Möglich, T., Kohn, N., Steiner, J.: Performance optimized communication system for high-dynamic and real-time robot control systems. In: Proceedings of 16th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2007), pp. 192–201 (2007)
20. Dadji, Y., Maass, J., Michalik, H.: Parallel task processing on a multicore platform in a pc-based control system for parallel kinematics. In: proceedings of the 6th International Conference on Computing, Communications and Control Technologies (CCCT), Orlando(FL), USA (2008)
21. Dadji, Y., Michalik, H., Möglich, T.: Parallel architecture for real time control system based on the ieee1394 standard. In: proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas(NV), USA (2009)
22. Beckmann, G.: Ein Hochgeschwindigkeits-Kommunikations-System für die industrielle Automation. Ph.D. thesis, Technische Universität Braunschweig (2001)
23. Beckmann, G., Varchmin, J.-U.: Industrial Automation Protocol - Kommunikationsprotokoll für die industrielle Automation auf Basis der Norm IEEE 1394, Version 0.9.5. Technical report, Institut für Elektrische Messtechnik und Grundlagen der Elektrotechnik, Technische Universität Braunschweig (2002)
24. Finkemeyer, B., Kröger, T., Kubus, D., Olschewski, M., Wahl, F.M.: Mirpa: Middleware for robotic and process control applications. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2007 (2007)
25. Ieee std. 1394-2008, standard for a high performance serial bus (2008)
26. Sercos 3 homepage (2009), `http://www.sercos.de/sercos-iii.175.0.html`
27. Macro.org - homepage (09.08.2006), `www.macro.org/`
28. Eberhardt, U., Kelm, H.J.: USB 2.0: Datendienste, Function, Hub, Host, Errorhandling, Powermanagement, USB-Treiber, USB-Bausteine, USB-Applikationen, Test & Analyse. Professional series. Franzis, Poing (2001) ISBN 3772379656
29. Breyer, R., Riley, S.: Switched, fast, and gigabit Ethernet: [understanding, building, and managing high-performance Ethernet networks], 3rd edn. Macmillan network architecture and development series. Macmillan Technical Publ., Indianapolis (1999) ISBN 1578700736
30. Iec 61158 (all parts), industrial communication networks - fieldbus specifications. Technical report
31. Prytz, G.: A performance analysis of ethercat and profinet irt. In: IEEE International Conference on Emerging Technologies and Factory Automation ETFA 2008, pp. 408–415 (2008)
32. Kohn, N.: Kommunikations-Infrastruktur für hochdynamische Parallelroboter. Ph.D. thesis, Technische Universität Carolo-Wilhelmina zu Braunschweig (2007)
33. Kohn, N., Varchmin, J.-U., Steiner, J., Goltz, U.: Universal communication architecture for high-dynamic robot systems using qnx. In: Proceedings of International Conference on Control, Automation, Robotics and Vision (ICARCV 8th), vol. 1, pp. 205–210. IEEE Computer Society, Kunming (2004) ISBN: 0-7803-8653-1
34. Steiner, J., Goltz, U., Maaß, J.: Self-Management within a Software Architecture for Parallel Kinematic Machines. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 355–371. Springer, Heidelberg (2010)
35. Finkemeyer, B., Kröger, T., Wahl, F.M.: A Middleware for High-Speed Distributed Real-Time Robotic Applications. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 193–212. Springer, Heidelberg (2010)

36. Diethers, C., Finkemeyer, B., Kohn, N.: Middleware zur Realisierung offener Steuerungssoftware für hochdynamische Prozesse. Information Technology (1), 39–47 (2004)
37. Hesselbach, J., Maaß, J., Bier, C.: Singularity prediction for parallel robots for improvement of sensor-integrated assembly. CIRP Annals - Manufacturing Technology 54(1), 349–352 (2005)
38. Bier, C., Maaß, J., Campos, A., Queiroz, E.: Direct singularity avoidance strategy for the hexa parallel robot. In: 18th International Congress of Mechanical Engineering (COBEM 2005), Ouro Preto, Brasilien (2005)
39. Hutchinson, S., Hager, G., Corke, P.: A tutorial on visual servo control. IEEE Transactions on Robotics and Automation 12(5), 651–670 (1996)
40. Anderson, D., Dzatko, D.: Universal Serial Bus system architecture, 2nd edn. PC system architecture series. Addison-Wesley, Boston (2001)
41. Dietrich, F., Maaß, J., Raatz, A., Hesselbach, J.: RCA562: Control Architecture for Parallel Kinematic Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 315–331. Springer, Heidelberg (2010)

Integrated Force and Motion Control of Parallel Robots – Part 1: Unconstrained Space

Michael Kolbus, Frank Wobbe, Thomas Reisinger, and Walter Schumacher

Abstract. Parallel robots in the context of handling and assembly benefit from their inherent high dynamics. However, their coupled nonlinear system dynamics demand for an effective control system in terms of accuracy, robustness and disturbance rejection. This paper presents a hybrid control scheme that is designed specifically to these features. The hierarchical approach encapsulates the robot dynamics on drive-level and, thus, provides the basis for effective real-time hybrid control. It is interfaced by a fully specified C^2-continuous trajectory. Moreover, concepts of virtual sensors are used for integration of additional components and mechanisms.

1 Introduction

As parallel robots begin to gain acceptance in industrial application, their control becomes more and more important. In contrast to their serial counterparts parallel kinematic structures combine a high dynamic potential with precision and stiffness. Consequently, control concepts have to address these features in order to maximize the performance of the robot. The control concept introduced in this paper focuses on parallel robots in the field of handling and assembly and is therefore optimized with respect to high dynamics. The control is embedded in the software infrastructure developed at the collaborative research center SFB 562. This infrastructure provides a complete functional chain from task generation to its execution on a physical layer. Additional demands for the control concepts are derived from the fact that tasks are specified by Manipulation Primitives, see [1].

Michael Kolbus · Frank Wobbe · Walter Schumacher
Technische Universität Braunschweig, Institute of Control Engineering,
Hans-Sommer-Straße 66, 38106 Braunschweig, Germany
e-mail: {kolbus,wobbe,w.schumacher}@ifr.ing.tu-bs.de

Thomas Reisinger
ABB Corporate Research Center, Wallstadter Straße 59, 68526 Ladenburg, Germany
e-mail: thomas.reisinger@de.abb.com

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 233–252.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

As presented in [2] the overall framework of the control architecture yields a highly modular design. The feedback control itself has to be modular to meet these concerns. Consequently, the control scheme presented here, is grouped into functional units. Thus, the approach in this paper provides the required flexibility and guarantees an extensible control structure that is well-suited for the demands of parallel robot control in operational space. A force-torque controller is integrated using the approach of [3].

Apart from the execution of hybrid control tasks, the controller integrates additional features. The concept of virtual sensors is included in the core unit and extended to support adaptronic components. The idea was inspired by the inclusion of adaptronic actors in the passive joints, see [4]. Their effect on the system dynamics is estimated and, thus, can be compensated. A similar approach is used to facilitate the change of assembly modes as presented in [5], where the controller has to operate in different state spaces to achieve this task. Both of the mentioned features are unique to parallel kinematic machines, since serial structures neither possess passive joints nor change their assembly modes.

The paper is organized as follows. Section 2 gives a short overview and classifies the presented control scheme in the context of the SFB 562. The following sections describe the concept in more detail, each one addressing a specific function. Section 3 discusses linearization and the subordinated drive controller. Additional features such as the change of assembly modes and integration of adaptronic components are pointed out in section 4. The last section summarizes the results and draws a conclusion.

2 Specification of Integrated Control

The design of an integrated control scheme is based on a torque driven interface to the inverters at the bottom layer and aims at tracking a trajectory specified by position, velocity and acceleration in the base frame of the robot. This guarantees encapsulation of robot dynamics and a uniform trajectory interface for the top layer.

Closed loop control operates on the torque interface of the inverters and thus provides manipulation on the robots dynamic behaviour. Due to the nonlinearities in the system dynamics and the demand for trajectory specification (in operational or joint space), an encapsulation scheme comes in quite handy. It establishes a higher level of abstraction and provides a robot-independent system dynamic by using feed forward or feedback-linearization-techniques and closed loop control of robot dynamics [6]. As a result, this layer of abstraction provides a linear (or more generally shaped) system dynamic, restricted only by the inverter dynamics. Hence, a top-layer interface is established by a fully specified C^2-continuous trajectory input $\{\boldsymbol{x}_{\mathrm{ref}}, \dot{\boldsymbol{x}}_{\mathrm{ref}}, \ddot{\boldsymbol{x}}_{\mathrm{ref}}\}$ specified with respect to the base frame of the robot (operational space). The robot dynamics are encapsulated giving a unified interface and enabling further control task definitions via this interface by motion modules.

3 Subordinated Drive Controller

Parallel structures come along with well-known nonlinear system dynamics, namely

$$\boldsymbol{M}_q(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_q(\dot{\boldsymbol{q}},\, \boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{\eta}_q(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{f}_{\mathrm{ext}} \tag{1}$$

in joint space (cf. used index q), which can be transfered to operational space (indicated by index x) to

$$\boldsymbol{M}_x(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{C}_x(\dot{\boldsymbol{q}},\, \boldsymbol{q})\dot{\boldsymbol{x}} + \boldsymbol{\eta}_x(\boldsymbol{q}) = \boldsymbol{G}\boldsymbol{\tau} + \boldsymbol{f}_{\mathrm{ext}} \tag{2}$$

by applying a coordinate transformation. Note, that $\boldsymbol{J}$ represents serial manipulator Jacobian and $\boldsymbol{G} = \boldsymbol{J}^{-\mathrm{T}}$ relates to the parallel manipulator Jacobian. In case of redundancy or underactuation only one representation contains the complete kinematic information, whereas the other one is obtained via the generalized inverse, cf. [7, 8].

To cope with the nonlinearities of the dynamic system eq. (2), the design approaches of the subordinated drive controller can be divided into two main groups: Linear control concepts based upon linearization techniques on the one hand and nonlinear ones such as sliding mode control on the other hand. Both concepts can be improved by the integration of observer based approaches for better disturbance rejection. Due to the chosen concept of design the performance of encapsulated dynamics is characterized as follows.

3.1 Preliminaries

Linearization techniques are classified into two groups and differ in selected values and signal flow for linearization: Exact feedback linearization on the one hand bases on measured values for model computation leading to a closed loop inverse dynamics control. Computed torque feed forward design on the other hand aims at linearization via reference values and feed forward. Besides, a special case will be presented in section 3.4, where a mix of both schemes is used for equivalent control.

In both cases, a precise model of the robot dynamics is vital to guarantee sufficient performance of linearization, cf. eqs. (1) and (2). In case of inverse dynamics feedback control an external reference input $\boldsymbol{u}$ is set up that renders the system as a set of double integrators for each axis in Cartesian space [9], by applying

$$\boldsymbol{\tau} = \boldsymbol{G}^{-1}\boldsymbol{M}_x\boldsymbol{u} + \boldsymbol{G}^{-1}\boldsymbol{\xi}_x, \quad \boldsymbol{\xi}_x = \boldsymbol{C}_x\dot{\boldsymbol{x}} + \boldsymbol{\eta}_x \tag{3}$$

as input for the torque-driven interface to the inverter.

Computed torque control on the contrary uses feed forward of reference values for linearization:

$$\boldsymbol{\tau} = \boldsymbol{G}^{-1}\boldsymbol{M}_x\ddot{\boldsymbol{x}}_{\mathrm{ref}} + \boldsymbol{G}^{-1}\boldsymbol{\xi}_{x,\mathrm{ref}} + \boldsymbol{M}_q\boldsymbol{v}, \quad \boldsymbol{\xi}_{x,\mathrm{ref}} = \boldsymbol{C}_x\dot{\boldsymbol{x}}_{\mathrm{ref}} + \boldsymbol{\eta}_x, \quad \boldsymbol{M}_q = \boldsymbol{G}^{-1}\boldsymbol{M}_x\boldsymbol{G}^{-\mathrm{T}}, \tag{4}$$

where $\boldsymbol{v}$ analogous to $\boldsymbol{u}$ in eq. (3) serves as the external input. Unlike inverse dynamics feedback control the obtained set of double integrators act in joint space

with respect to input $\boldsymbol{v}$. Note that eq. (4) takes advantage of matrix dependence on joint variables in eq. (2) and thus circumvents calculation of the direct kinematic problem (DKP).

Due to the large differences between the electrical and mechanical time constants, the dynamic effects of the inverters within the linearization have been neglected up to this point. However, the delay of the inverters affects the described linearization. Instead of a set of double integrators, feedback (eq. (3)) and feed forward linearization (eq. (4)) result in

$$V_\theta^{\langle \mathrm{k} \rangle} u^{\langle \mathrm{k} \rangle} = T_{\mathrm{el}} \dddot{x}^{\langle \mathrm{k} \rangle} + \ddot{x}^{\langle \mathrm{k} \rangle} \quad \text{and} \quad V_\theta^{\langle \mathrm{k} \rangle} v^{\langle \mathrm{k} \rangle} = T_{\mathrm{el}} \dddot{q}_k + \ddot{q}_k, \qquad k \in \{1, \ldots, n_q\} \tag{5}$$

for each degree-of-freedom $\langle \mathrm{k} \rangle$, respectively.

In eq. (5) $V_\theta^{\langle \mathrm{k} \rangle}$ represents the gain due to inertia, which is considered as $T_v = \frac{1}{V_\theta}$ during parametric design. The number of degrees of freedom and delay of the inverters are denoted by n_q and T_{el}, respectively. In case of model uncertainties $V_\theta^{\langle \mathrm{k} \rangle} \neq 1$ is valid. Moreover, neglected nonlinear terms in eq. (5) have to be considered as disturbances for the design of the top layer axis controller in the next section. Comparing both schemes it can be seen that feedback linearization results in control in operational space, e. g. centralized control, whereas feed forward linearization leads to decentralized control in joint space. Taking into account, that for parallel robots the inverse kinematics problem (IKP) in general is easier to solve than the direct kinematic one (DKP), the use of computed torque feed forward linearization is more convenient.

3.2 Classical Approach

Linear cascaded control concepts can be applied for closed loop control by using linearization techniques (cf. eqs. (3) and (4)). Due to different linearization types the overlying control scheme is denoted as centralized control in case of exact linearization and decentralized or computed torque control for feed forward linearization, cf. [8]. Both schemes feature a cascaded structure (cf. Fig. 1 and 2, respectively) where the elements can be described via transfer functions:

$$K_v(s) = V_1 \frac{T_I s + 1}{T_I s}, \quad K_p(s) = V_2 \frac{T_R s + 1}{T_L s + 1}. \tag{6}$$

In the view of control two major design goals are considerable: Trajectory tracking on the one hand and counteracting model uncertainties on the other hand. Taking these specifications into account, the design is optimized for disturbance rejection and feed forward dynamics.

Parameters are determined by comparing the denominator of the closed loop dynamics with a model function. Confining all poles except one complex pole pair to the real axis allows the extraction of damping ratio D as an independent parameter. Using an integral criterion for minimizing the disturbance step response yields optimized parameters as

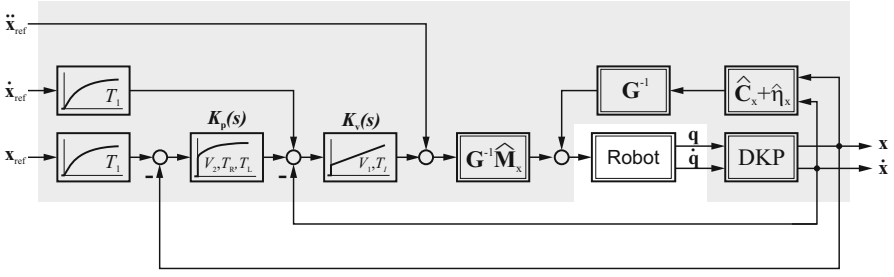


Fig. 1. Centralized control based on exact feedback linearization

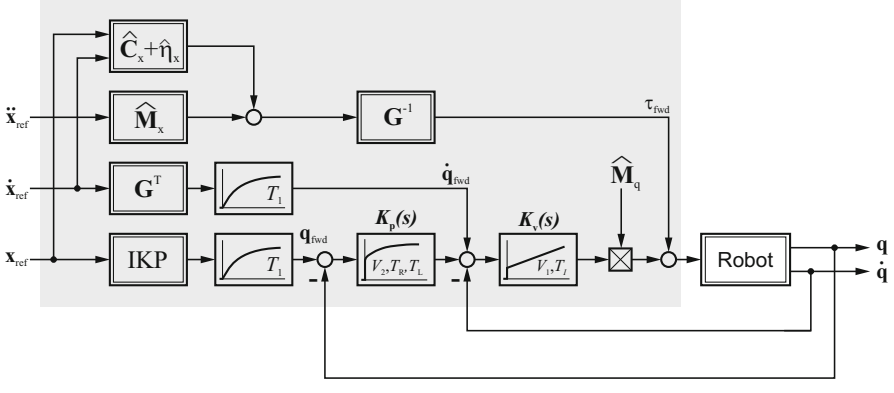


Fig. 2. Decentralized control based on feed forward linearization

$$V_1 = \frac{\left(5D^2+1\right)T_v}{16D^2T_{\mathrm{el}}}, \qquad T_I = \frac{4T_{\mathrm{el}}\left(5D^2+1\right)}{1+2D^2},$$
$$V_2 = \frac{1}{4T_{\mathrm{el}}\left(1+2D^2\right)}, \quad T_R = 4T_{\mathrm{el}}, \qquad T_L = T_I. \tag{7}$$

For detailed design consider [10].

3.3 Observer Based Approaches

For disturbance rejection structural design patterns are inevitable. Although the design in the former section guarantees disturbance rejection by parametric design, an explicit addressing in structural design is quite handy and results in well-known observer based structures. The core idea herein is to observe disturbances and feed them forward. Among various concepts, the principle of input balancing as introduced in [11] stands out, as it combines tracking and disturbance rejection – a direct feedthrough in feed forward control is used and amended by a disturbance observer. Thus, command action is improved by direct feedthrough, whereas

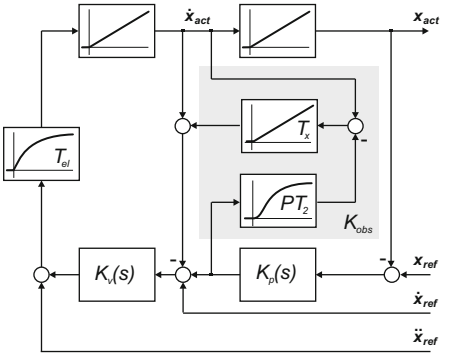


Fig. 3. Block diagram Input balancing

robustness against external disturbances is assured by using a model of the velocity loop for disturbance observation, which contrasts with classical observer design. The overall control structure relies on linearization techniques and is presented in Fig. 3 for the case of centralized control. The parametric design is based on cascade control with damping ratios D_v and D_p for velocity and position plant, respectively, and applies the control laws:

$$K_v(s) = V_1, \quad K_p(s) = V_2, \quad K_{PT_2}(s) = \frac{1}{(\frac{s}{\omega_0})^2 + \frac{2D_v}{\omega_0}s + 1}, \quad K_x(s) = \frac{1}{T_x s} \tag{8}$$

with $K_{PT_2}(s)$ as model of the closed loop velocity cascade. Disturbances are here modeled and matched by an integrator $K_x(s)$. Parameters result in

$$V_1 = \frac{T_v}{3T_{\mathrm{el}}}, \quad V_2 = \frac{1}{9T_{\mathrm{el}}}, \quad \left(\frac{s}{\omega_0}\right)^2 = 3T_{\mathrm{el}}^2, \quad \frac{2D_v}{\omega_0} = 3T_{\mathrm{el}}, \quad T_x = 9T_{\mathrm{el}} \tag{9}$$

when selecting damping ratios of $D_v = \frac{\sqrt{3}}{2}$ and $D_p = 1$ for the velocity and position cascade, respectively. Note, that $D_v = \frac{\sqrt{3}}{2}$ equals an damping ratio of 1 for input of disturbances in the velocity plant. For a detailed parametric design see [12]. Further optimization can be achieved by not only considering steplike but also dynamic disturbances – which is ensured by stepping from an integrator- to an *PI*-model for disturbances. A more detailed discussion of this concept can be found in [10] and [12]. An improvement in trajectory tracking can be expected, however, model uncertainties nonetheless have impact on the dynamic behavior.

3.4 Sliding Mode Approach

Model uncertainties have not yet been explicitly considered in the design process. This can be accomplished by nonlinear control concepts such as sliding mode. The basic concept has been discussed by Utkin [13] and was taken up by Slotine [14]

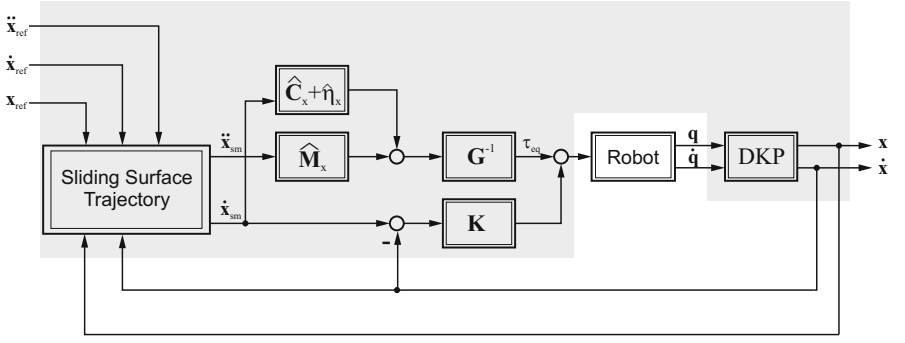


Fig. 4. Sliding mode control with continuous sliding surface

with a general definition of sliding surfaces and boundary layers to lessen the effect of chattering. It focuses on the dynamics of the tracking error and binds it to a first order dynamic by defining a sliding surface

$$\boldsymbol{s} = \dot{\widetilde{\boldsymbol{x}}} + \boldsymbol{\Lambda}\widetilde{\boldsymbol{x}}, \qquad \widetilde{\boldsymbol{x}} = \boldsymbol{x}_{\mathrm{act}} - \boldsymbol{x}_{\mathrm{ref}}, \tag{10}$$

where $\boldsymbol{\Lambda}$ is a positive definite matrix. To bind the error to the sliding surface $\boldsymbol{s} = \boldsymbol{0}$ the reference trajectory is replaced by a virtual one $\{\boldsymbol{x}_{\mathrm{sm}}, \dot{\boldsymbol{x}}_{\mathrm{sm}}, \ddot{\boldsymbol{x}}_{\mathrm{sm}}\}$ where

$$\boldsymbol{x}_{\mathrm{sm}} = \boldsymbol{x}_{\mathrm{ref}} - \boldsymbol{\Lambda} \int_0^t \widetilde{\boldsymbol{x}} \mathrm{d}t. \tag{11}$$

Using eqs. (11) and the equivalent dynamics set point $\boldsymbol{\tau}_{\mathrm{eq}}$ in Fillipow's sense [15] the control law

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{\mathrm{eq}} - \boldsymbol{w} = \boldsymbol{G}^{-1}\left(\widehat{\boldsymbol{M}}_x \ddot{\boldsymbol{x}}_{\mathrm{sm}} + \widehat{\boldsymbol{C}}_x \dot{\boldsymbol{x}}_{\mathrm{sm}} + \widehat{\boldsymbol{\eta}}_x\right) - \boldsymbol{K}\boldsymbol{s} \tag{12}$$

ensures stability and is depicted in Fig.4. The additional input $\boldsymbol{w}$ counteracts model uncertainties and guarantees precise tracking [16]. In contrast to classical sliding mode approaches the chattering formally associated with sliding mode control is obliterated by the continuous sliding surface and abdication of discontinuities such as switching terms. As a result chattering is minimized at the price of slightly (i. e. from theoretical point of view) reduced performance compared to variable structure control. The parametric design mainly depends on the matrix $\boldsymbol{\Lambda}$ as it resembles the first order error dynamics and is obtained by

$$\boldsymbol{\Lambda} = \frac{1}{3T_{\mathrm{el}}}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \boldsymbol{K} = \boldsymbol{G}^{-1}\widehat{\boldsymbol{M}}_x\boldsymbol{\Lambda}. \tag{13}$$

To explicitly force integral controller action, the sliding surface eq. (10) can be redefined with an integral of tracking error

$$\boldsymbol{s} = \dot{\tilde{\boldsymbol{x}}} + 2\boldsymbol{\Lambda}\tilde{\boldsymbol{x}} + \boldsymbol{\Lambda}^2 \int_0^t \tilde{\boldsymbol{x}} \mathrm{d}t. \tag{14}$$

Thus, integral action can be used to enhance disturbance rejection. For extensions to higher order sliding modes we refer to the works presented in [17].

3.5 Comparison

Parametric designs presented in sections 3.2, 3.3 and 3.4 feature different characteristics, which are analyzed based on closed loop dynamics. The focus is set on performance of feed forward dynamics and robustness against parameter variation.

Referring to the first one, feed forward dynamic, performance analysis has to be separated into two aspects: linearization technique and equivalent linear system closed loop dynamics. For sake of simplicity the analysis of linearization technique is kept rather short – it can be found widely in literature [18, 19, 20]. In short, weak points of each linearization technique are influence of measurement noise for centralized control, drift of linearization in case of trajectory following error in decentralized control and both points, however, to a far lesser extent, for equivalent control (as present in sliding mode control). The closed loop system dynamics of an equivalent linear system are depicted in Table 1. For sliding mode control the nonlinear system dynamics are given in Table 2, where for sake of simplicity inverter dynamics have been neglected. A consideration can be found in [17], revealing that similar dynamics with sliding of order two are obtained.

Towards performance comparison of the presented control schemes it can be seen that input balancing offers best bandwidth for tracking purposes. On the one hand, static disturbances are rejected by all presented control schemes; the optimized cascade control is just slightly outperformed by input balancing. Sliding mode control on the other hand is marked by nonlinear closed loop dynamics and consists of two parts: In case of disturbances, parametric errors and other uncertainties it is described by the reaching phase towards the sliding surface which mainly depends on $\boldsymbol{K}$. Considering eq. (13), a time constant of $\boldsymbol{\Lambda}^{-1}$ can be noted. When attaining the sliding surface $\boldsymbol{s} = \boldsymbol{0}$, the system enters sliding mode. The system output error $\boldsymbol{x}$ converges exponentially to zero (with time-constant $\boldsymbol{\Lambda}^{-1}$ for classical sliding mode eq. (10) and $2\boldsymbol{\Lambda}^{-1}$ when using integral action eq. (14)).

Table 1. Closed loop system dynamics – disturbance and feed forward transfer functions

	Feed Forward	Disturbance
Cascade (optimized)	$\frac{1}{(4T_{\mathrm{el}}s+1)^3}$	$\frac{256T_{\mathrm{el}}^3 s(T_{\mathrm{el}}s+1)}{(4T_{\mathrm{el}}s+1)^4}$
Input balancing	$\frac{1}{(3T_{\mathrm{el}}s+1)^3}$	$\frac{243T_{\mathrm{el}}^3 s(T_{\mathrm{el}}s+1)(3T_{\mathrm{el}}^2s^2+3T_{\mathrm{el}}s+1)}{(3T_{\mathrm{el}}s+1)^6}$

Table 2. Closed loop system dynamics of sliding mode control – error and disturbance dynamics

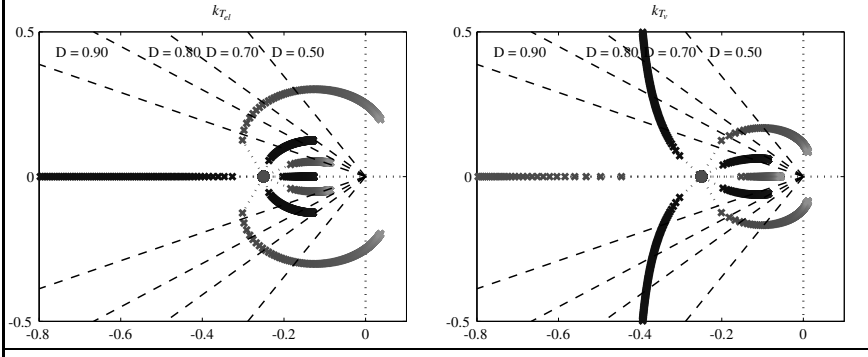
	Error & Sliding dynamics	Disturbance dynamics
Classical	$\mathbf{0} = \boldsymbol{M}_x\dot{\boldsymbol{s}} + (\boldsymbol{C} + \boldsymbol{GK})\,\boldsymbol{s}$ $\mathbf{0} = \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{x}}$	$\boldsymbol{M}_x\ddot{\tilde{\boldsymbol{x}}} + (2\boldsymbol{M}_x\boldsymbol{\Lambda} + \boldsymbol{C}_x)\dot{\tilde{\boldsymbol{x}}} + (\boldsymbol{M}_x\boldsymbol{\Lambda} + \boldsymbol{C}_x)\boldsymbol{\Lambda}\tilde{\boldsymbol{x}} = \boldsymbol{d}$
Integral action	$\mathbf{0} = \boldsymbol{M}_x\dot{\boldsymbol{s}} + (\boldsymbol{C} + \boldsymbol{GK})\,\boldsymbol{s}$ $\mathbf{0} = \ddot{\tilde{\boldsymbol{x}}} + 2\boldsymbol{\Lambda}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{\Lambda}^2\int_0^t \tilde{\boldsymbol{x}}\,\mathrm{d}t$	$\boldsymbol{M}_x\dddot{\tilde{\boldsymbol{x}}} + (3\boldsymbol{M}_x\boldsymbol{\Lambda} + \boldsymbol{C}_x)\ddot{\tilde{\boldsymbol{x}}} + (3\boldsymbol{M}_x\boldsymbol{\Lambda} + 2\boldsymbol{C}_x)\boldsymbol{\Lambda}\dot{\tilde{\boldsymbol{x}}} + (\boldsymbol{M}_x\boldsymbol{\Lambda} + \boldsymbol{C}_x)\boldsymbol{\Lambda}^2\tilde{\boldsymbol{x}} = \dot{\boldsymbol{d}}$

Robustness against parameter variation is validated for dominant system parameters by design – the inertia of the mechanical System $V_\theta = \frac{1}{T_v}$ and the delay introduced by the inverter and communication T_{el}. Although the virtual inertia is compensated by linearization and thus equals 1, it varies in case of model uncertainties and payload changes. It is considered by an multiplicative uncertainty $k_{T_v} = \frac{T_v'}{T_v}$, where the prime denotes actual system parameters whereas the other variable indicates assumed parameter for controller design. The same considerations apply to the time delay T_{el}, which cannot match the inverter dynamics and delay in communication perfectly and thus contains uncertainty. Like the variation of the inertia it is represented by a factor $k_{T_{\mathrm{el}}} = \frac{T_{\mathrm{el}}'}{T_{\mathrm{el}}}$. Reformulation of closed loop dynamics for optimized cascade and input balancing and, by using parameters eqs. (7) and (9) respectively with respect to parameter uncertainties k_{T_v} and $k_{V_{\mathrm{el}}}$, results in transfer functions presented in Table 3. Impact of parameter mismatch can be seen in parametric plots depicted in Table 3 where lower brightness indicates a smaller parametric mismatch. Within the range of parameter variation it can be seen that both control schemes are more sensitive to variations when parameters are assumed smaller than reality, i. e. $k_{T_i} > 1$, cf. light gray branches. Both parameters have significant impact on closed loop system dynamics, with input balancing revealing more sensitivity towards parameter uncertainties compared to cascade control. With the same step size and range of parameter mismatch input balancing reveals two weak points: Firstly, small perturbations of parameters have deeper impact on closed loop system damping as pointed out in figures of Table 3. Secondly stability is diminished to the same amount by less parametric mismatch compared to cascade control as manifested by Fig. 5.

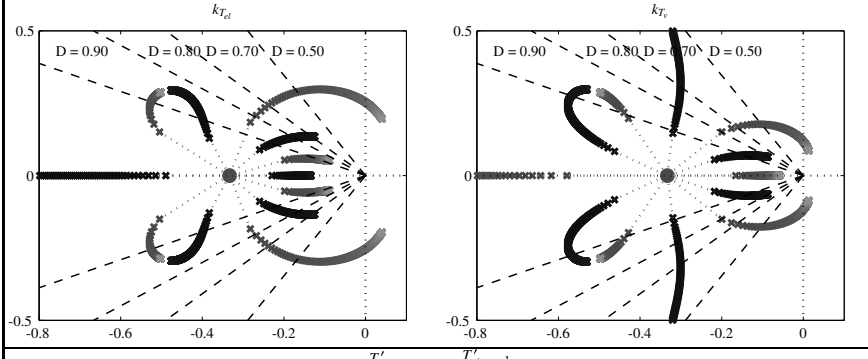
Thus, its better bandwidth for command following is traded off by lesser robustness against parameter uncertainties. As handling and assembly in the field of robotics has to cope with changes of payload, parametric design of optimized cascade control is preferable.

Table 3. Closed loop system dynamics – influence of model uncertainties. Top: cascade control, bottom: input balancing. The branches in dark gray indicate smaller system parameters than assumed for controller design, branches in light gray the opposite. Lower brightness within a branch indicates a smaller parametric mismatch.

$$G(s) = \frac{a_\mathrm{C} + 1}{k_{T_\mathrm{el}} k_{T_v} a_\mathrm{C}^4 + 4 k_{T_v} a_\mathrm{C}^3 + 6 a_\mathrm{C}^2 + 4 a_\mathrm{C} + 1}$$



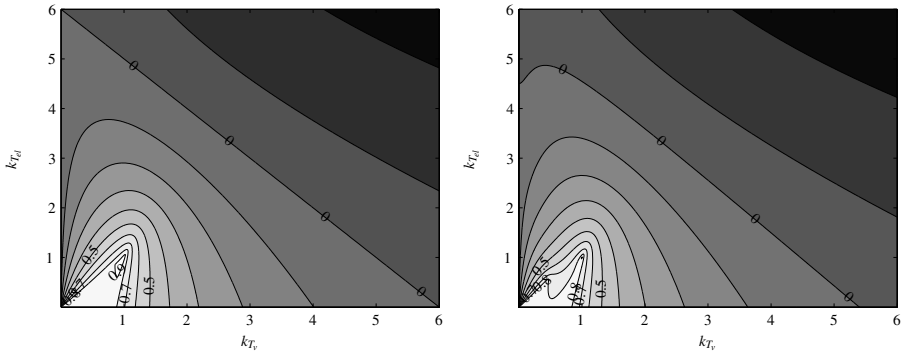
$$G(s) = \frac{a_\mathrm{I}^3 + 3a_\mathrm{I}^2 + 3a_\mathrm{I} + 1}{k_{T_\mathrm{el}} k_{T_v} a_\mathrm{I}^6 + 3 k_{T_v} \left(1 + k_{T_\mathrm{el}}\right) a_\mathrm{I}^5 + 3 \left(1 + k_{T_\mathrm{el}} k_{T_v} + 3 k_{T_v}\right) a_\mathrm{I}^4 + \left(9 k_{T_v} + 11\right) a_\mathrm{I}^3 + 15 a_\mathrm{I}^2 + 6 a_\mathrm{I} + 1}$$



$a_\mathrm{C} = 4T_\mathrm{el}s$, $a_\mathrm{I} = 3T_\mathrm{el}s$, $k_{T_v} = \frac{T'_v}{T_v}$, $k_{T_\mathrm{el}} = \frac{T'_\mathrm{el}}{T_\mathrm{el}}$, $\frac{1}{10} < k_i < 1$, $k_i = 1$, $1 < k_i < 10$

T'_i: real system parameter, T_i: assumed parameter for controller design

A full analysis of parametric mismatch on the underlying linearization scheme has not been considered in detail so far, as the effect of a mismatch is handled by the linear control scheme. Sliding mode control in contrast to these classical design concepts, however, addresses parameter uncertainties in the design process directly and, thus, is more robust. To give a short comparison to classical linear design a variation of mass with $\boldsymbol{M}'_x = k_{T_v} \boldsymbol{M}_x$ analogues to Table 3 is analyzed. For convergence and stability the sliding condition $\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{s}^2 \leq -\nu|\boldsymbol{s}|$ must hold and is used for performance analysis under parameter mismatch. Further analysis reveals



(a) Cascade control (b) Input balancing

Fig. 5. Parametric sensitivity - smallest damping ratio of closed loop system dynamics. Level curve at zero indicates the stability limit.

$$\dot{\boldsymbol{s}} = -\left(k_{T_v}\boldsymbol{M}_x\right)^{-1}\left(\boldsymbol{C}_x + \boldsymbol{G}\boldsymbol{K}\right)\boldsymbol{s} - \frac{1-k_{T_v}}{k_{T_v}}\left(-\ddot{\boldsymbol{x}}_{\mathrm{ref}} + 2\Lambda\dot{\tilde{\boldsymbol{x}}} + \Lambda^2\tilde{\boldsymbol{x}}\right). \tag{15}$$

In case of exact parameter match $k_{T_v} = 1$ the last term in eq. (15) vanishes and thus, the sliding condition is guaranteed. Again, assuming parameters smaller than reality, i. e. $k_{T_i} > 1$, is critical as the settling time increases. It can be seen, that performance not only depends on parameter variation, but also on trajectory dynamics, which is different from a linear design. For exact results, an analysis specific to each individual setup has to be considered and is skipped for sake of simplicity.

Taking further model uncertainties $\widetilde{\boldsymbol{M}}_x = \widehat{\boldsymbol{M}}_x - \boldsymbol{M}_x$, $\widetilde{\boldsymbol{C}}_x = \widehat{\boldsymbol{C}}_x - \boldsymbol{C}_x$ and $\widetilde{\boldsymbol{\eta}}_x = \widehat{\boldsymbol{\eta}}_x - \boldsymbol{\eta}_x$ into account and considering parametric design eq. (13) it can be seen that sliding mode offers robustness against parameter uncertainties in case of integral action. Moreover, it is more robust than control schemes based upon linearization techniques. For a more detailed analysis we refer to [21]. Compared to results obtained via control based upon linearized subsystem sliding mode offers more robustness against parametric uncertainty.

Experimental setup for given parametric designs and validations on the planar parallel robot FIVEBAR can be found in [6]. As a summary it can be noted that sliding mode turns out to be a promising alternative to classical linear control concepts on parallel robots. Instead of considering nonlinearities on the innermost level it addresses inclusion of uncertainties in an overall design. The demand for a fully specified trajectory is no restriction as jerk limitation has to be taken into account in design of motion modules for preventing structural damage. Nonetheless, centralized and decentralized control feature certain advantages, namely operational path accuracy and best suppression of noisy velocity signals as well as no need for solving the DKP. As has been pointed out in the parametric design choices, the optimized cascaded structure offers advantages with respect to parametric uncertainties and corresponding range of stability, thus providing an optimal control scheme in terms of accuracy, robustness and disturbance rejection. Hence, within the SFB 562

computed torque feed forward control with parametric design of optimized cascaded structure is favored. Experimental results of this setup on TRIGLIDE and HEXA are presented in [3], proving the performance and integration into the uniform hierarchical framework. As a result robot dynamics are encapsulated and a top-layer interface is provided with a fully specified C^2-continuous trajectory input interface.

4 Integration of Adaptronic Components and Change of Workspaces

High performance of robots in the field of handling and assembly demands for additional features in order to extend the capabilities of the robot. Additional components and concepts investigated in the SFB 562 are presented in [4]. They are used to overcome common disadvantages of parallel robots, cf. [5], and increase the versatility of the robot as proposed in [4]. Two concepts are introduced in this chapter: Virtual sensors for the integration of adaptronic components and a control scheme that supports the change of assembly modes.

The concept of virtual sensors allows the integration of adaptronic joints. These sensors provide the necessary state space information without the cost of additional hardware. They are instantiated by nonlinear state observers. Thus, the derivation of the variables depends on existing measurement combined with a precise process model.

4.1 Change of Assembly Modes

One major weakness of parallel robots is their poor ratio of installation to workspace. Its optimization can be addressed during the design of the robot, but it remains an inherent problem. An approach to increase the ratio is presented in [22]. By changing the configuration of the robot several adjacent workspaces can be utilized. The combined workspace can be significantly larger and thus, the afore mentioned ratio of installation to workspace improves. For example the experimental robot TRIGLIDE almost doubles its workspace.

Switching configuration and thus, workspaces requires concepts for passing through singularities. As shown in [22] the structure of the TRIGLIDE has to pass 3 singularities of type 1 and an additional type 2 singularity. The approach presented here uses gravity for safe passing of the singularity of type 2. In terms of control engineering the corresponding axis has to be force controlled on the drive level during the process. The trajectory interface is not suitable for this task, and therefore, configuration change has to be handled on the drive level.

Unfortunately, the hardware only provides a velocity interface to the robot. Therefore, the velocity controller has to be included in the design. This leads to a linearized model of the axis while passing the singularity. Fig. 6 shows the system model, where F_{ext} represents the gravitational force that drives the system. The output of the velocity controller is treated as disturbance that has to be compensated. The target motion, i. e. passing the singularity of type 2, is carried out, if $F_m = 0$ and,

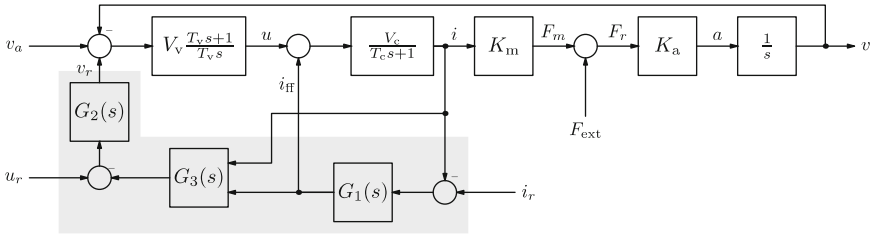


Fig. 6. Structure of the system and control of the drive while passing the singularity of type 2

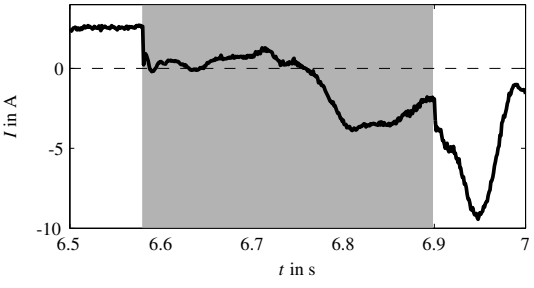


Fig. 7. Current of the relevant drive during configuration change. The gray region highlights the phase where the velocity controller is suppressed.

thus, $u = 0$ holds. Due to different sample times of the internal velocity controller and other parts of the control system, positive feedback of the velocity signal is not sufficient for compensation. Hence, an observer for force and closed loop control is used.

The design of the controller is carried out in a linear framework. The systems inputs are identified to current feed-forward i_{ff} and velocity command v_r, whereas the output is the actual current i. For design, the controller is split into three modules G_i with $i = 1, 2, 3$, see Fig. 6. The inner loop is closed by G_1, whereas the overruling of the velocity controller is implemented by a disturbance observer. The output of the observer is the actuating variable of the velocity controller u, which otherwise is not accessible. Using this information, a synthetic additional velocity signal is generated that drives the output of the velocity controller to zero. In presence of noise the quality of the estimated signal can be improved by the use of a Kalman filter.

Experiments on the TRIGLIDE were carried out to evaluate the control scheme. Fig. 7 shows the current of the switched passive drive during the configuration change. The residual force in Fig. 7 dampens the initiated motion and will not cause unstable behaviour. The robot safely passes the singularity of type 2 by gravity.

4.2 Integration of Adaptronic Components

New bearings have been designed, which include adaptronic components to adjust the bearing clearance during operation to increase overall precision, see [23]. These adaptronic components have to be controlled to compensate for wear in the bearing

mechanics. The optimal bearing clearance is defined by the amount of friction involved. The variation of friction in the joints changes the dynamic properties of the robot.

Moreover, if the joints are not used in binary states, i. e. switched on or off, the current state of the joints needs to be determined for feedback. Sensors for measuring clearance or friction inside bearings do not exist. However, the influence of the adaptronic joints on the dynamic behaviour of the system allows the estimation of the afore mentioned variables using a state-observer approach. Thus, virtual sensors for these state variables are provided. Their output can be used for control and, thus, increase the performance by compensation or computation of trajectories with modified dynamics.

State estimation by observers in a linear framework is comprehensively covered in literature, whereas the nonlinear case lacks a unified approach. Differential geometry is a valuable tool for mechanic systems. Transformation of coordinates offers different views onto the system and, if possible, provides the possibility of canonical forms, which simplifies the design of observers.

A first evaluation of this concept has been done by a single axis implementation, see [24]. The results are used to compensate the friction torque, which lead to an improved performance. Estimation of friction in multiple joints is outlined in the following section.

4.2.1 Observability of the System

The first step in the design of an observer is the proof of the observability of the system. Without loss of generality the system is analyzed in generalized coordinates $\boldsymbol{q}$ to avoid unnecessary transformations. The dynamics of the robot are given in the following form (cf. eq. (1))

$$\boldsymbol{M}_q\ddot{\boldsymbol{q}} + \boldsymbol{C}_q\dot{\boldsymbol{q}} + \boldsymbol{\eta}_q = \boldsymbol{\tau} + \boldsymbol{J}_q^{\mathrm{T}}\boldsymbol{\tau}_f, \tag{16}$$

where $\boldsymbol{J}_q$ denotes the Jacobian to passive joints. The index q indicates that the matrix is derived in generalized coordinates. The robot is assumed to move in unconstrained space.

For observer design the friction torques have to be included as state variables in the dynamic equations of the system. The selected approach assumes Coulomb friction, such that

$$\boldsymbol{\tau}_f = \begin{bmatrix} -\sigma_1 \operatorname{sign}\left(\dot{\beta}_1\right) \\ \vdots \\ -\sigma_n \operatorname{sign}\left(\dot{\beta}_n\right) \end{bmatrix} = -\operatorname{diag}\left(\sigma_1, \ldots, \sigma_n\right) \cdot \operatorname{sign}\left(\dot{\boldsymbol{\beta}}\right) = -\operatorname{diag}\left(\boldsymbol{\sigma}\right) \cdot \dot{\boldsymbol{\beta}}_e, \tag{17}$$

where $\dot{\boldsymbol{\beta}}_e = \operatorname{sign}\left(\dot{\boldsymbol{\beta}}\right)$ and $\dot{\boldsymbol{\beta}}$ denote the velocity of the passive joints. The parameter σ_i combines the effect of normal force and coefficient of friction. Using eqs. (16) and (17) the system is transformed into a nonlinear state space representation

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\\ \boldsymbol{x}_3\end{bmatrix} = \begin{bmatrix}\boldsymbol{x}_2\\ \boldsymbol{M}_q^{-1}\left(-\boldsymbol{J}_q^{\mathrm{T}}\,\mathrm{diag}(\boldsymbol{x}_3)\dot{\boldsymbol{\beta}}_e - \boldsymbol{C}_q\boldsymbol{x}_2 - \boldsymbol{\eta}_q\right)\\ \boldsymbol{0}\end{bmatrix} + \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{M}_q^{-1}\boldsymbol{\tau}\\ \boldsymbol{0}\end{bmatrix} \tag{18a}$$

$$\boldsymbol{y} = \boldsymbol{h}_x = [\boldsymbol{x}_1, \boldsymbol{x}_2]^{\mathrm{T}}\,. \tag{18b}$$

The variables are renamed to $\boldsymbol{x} = [\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\sigma}]^{\mathrm{T}}$ and the output is set to position and velocity on drive level. The order of the system of differential equations is $n_s = 3n$, where n is the number of degrees of freedom of the robot.

The nonlinear state space representation allows the derivation of the observability map in a formalized way. It is constructed from time derivatives of the output map of the system up to the order of $n_s - 1$, see [25], and results in

$$\boldsymbol{q}_h^{\mathrm{T}} := \left[\frac{\mathrm{d}^0}{\mathrm{d}t^0}\boldsymbol{y}, \ldots, \frac{\mathrm{d}^{(n_s-1)}}{\mathrm{d}t^{(n_s-1)}}\boldsymbol{y}\right] = {}^{[n_s-1]}\boldsymbol{y}\,. \tag{19}$$

The observability map depends on the state variables, the input and its derivatives up to degree $n_s - 1$. Following the theorem stated in [25], the observable subspace of the system can be derived from the map eq. (19). The system is observable on the domain $\mathcal{D}_x$, if and only if

$$\forall \boldsymbol{x} \in \mathcal{D}_x : \quad \exists\, \boldsymbol{x} = \boldsymbol{q}_h^{-1}, \tag{20}$$

which declares the observability map injective on $\mathcal{D}_x$. Thus, it has a unique solution for $\boldsymbol{x}$. For systems with multiple outputs eq. (20) may have a unique solution using less than $(n_s - 1)$ derivatives.

Computing the derivatives of the output map specified in system eq. (18) leads to

$$\boldsymbol{y} = \begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}, \quad \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{y} = \begin{bmatrix}\boldsymbol{x}_2\\ \boldsymbol{M}_q^{-1}\left(\boldsymbol{\tau} - \boldsymbol{J}_q^{\mathrm{T}}\,\mathrm{diag}(\boldsymbol{x}_3)\dot{\boldsymbol{\beta}}_e - \boldsymbol{C}_q\boldsymbol{x}_2 - \boldsymbol{\eta}_q\right)\end{bmatrix} \tag{21}$$

$$\text{with} \quad \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{\tau}) = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} + \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{\tau}}\dot{\boldsymbol{\tau}}.$$

The observability map is given by eq. (21) to $\boldsymbol{q}_h = [\boldsymbol{y}, \dot{\boldsymbol{y}}]^{\mathrm{T}}$. Inversion with respect to $\boldsymbol{x}_1, \boldsymbol{x}_2$ is trivial, whereas explicit calculation of $\boldsymbol{x}_3$ demands for a solution of

$$\boldsymbol{J}^{\mathrm{T}}\,\mathrm{diag}(\dot{\boldsymbol{\beta}}_e)\boldsymbol{x}_3 = \boldsymbol{\tau} - \boldsymbol{\eta}_q - \boldsymbol{C}_q\boldsymbol{q}_{h,2} - \boldsymbol{M}_q\boldsymbol{q}_{h,4} \quad \text{with} \quad \boldsymbol{q}_{h,i} = [\boldsymbol{q}_{h,1}, \ldots, \boldsymbol{q}_{h,4}]^{\mathrm{T}}. \tag{22}$$

Hence the matrix $\boldsymbol{J}^{\mathrm{T}}\,\mathrm{diag}(\dot{\boldsymbol{\beta}}_e)$ has to be non-singular, i. e. each factor must have full rank:

$$\mathrm{rank}\left(\mathrm{diag}(\dot{\boldsymbol{\beta}}_e)\right) = n \quad \Rightarrow \quad \forall i \in \{1, \ldots, n\} : \quad \dot{\beta}_i \neq 0, \tag{23a}$$

$$\mathrm{rank}\left(\boldsymbol{J}^{\mathrm{T}}\right) = n \quad \Rightarrow \quad \text{non-singular position w. r. t. passive angles.} \tag{23b}$$

Computing further derivatives of the output map does not extend the observable domain. Therefore, Coulomb friction is only observable for non-zero passive

velocities and non-singular positions of the robot. The matrix needs to be left invertible for $\boldsymbol{q}_h$ as an injective map. This also limits the maximum number of friction torques that can be estimated to $n_{\mathrm{f,max}} = n$. Considering eq. (21) it can also be concluded that the observability of the system does not depend on the accessibility of angular velocity.

4.2.2 Derivation of an Observer

Having proven the observability of the system, an observer to estimate the variables of interest can be designed. High-gain observers are a common approach for mechanical systems and thus, can be applied to robotic manipulators [26]. The system needs to be expressed in observability canonical form. The transformation for control affine systems is given by the diffeomorphism,

$$\boldsymbol{t}_{\mathrm{obcf}} := \left[t_{\mathrm{obcf},0}, \ldots, t_{\mathrm{obcf},n_s-1}\right]^{\mathrm{T}}, \quad t_{\mathrm{obcf},k} := \mathrm{L}_f^k(h)(x), \tag{24}$$

see [27]. The differential operator $\mathrm{L}_f(h)(x)$ is identical to the Lie-derivative for systems without input.

The diffeomorphism eq. (24) is related to the observability map eq. (19) and renders the system to a chain of integrators with isolated nonlinearity

$$\begin{aligned} \dot{\boldsymbol{z}}_{\mathrm{obcf}} &= \dot{\boldsymbol{t}}_{\mathrm{obcf}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ 0 & \boldsymbol{0}^{\mathrm{T}} \end{bmatrix} \boldsymbol{z}_{\mathrm{obcf}} + \begin{bmatrix} \boldsymbol{0} \\ f_{\mathrm{obcf}}(\boldsymbol{z}_{\mathrm{obcf}}, \boldsymbol{u}) \end{bmatrix} = \boldsymbol{A}\boldsymbol{z}_{\mathrm{obcf}} + \boldsymbol{f}_{\mathrm{obcf}}, \\ h_{\mathrm{obcf}} &= z_{\mathrm{obcf},1}. \end{aligned} \tag{25}$$

The extension to multiple output systems is carried out by stacking the single output systems. The observer based on the new coordinates is given by

$$\dot{\hat{\boldsymbol{z}}}_{\mathrm{obcf}} = \boldsymbol{A}\hat{\boldsymbol{z}} + \boldsymbol{p}(y - \hat{z}_q), \quad p_i = p_i^* / \varepsilon^i, \tag{26}$$

with $P(s) = \sum_{i=0}^{q} p_i s^{q-i}$ being Hurwitz and ε a sufficiently small constant.

The above scheme lacks the inclusion of the model. Furthermore, its stability relies on high gains (i. e. small ε) to suppress the effects of the nonlinear term $\boldsymbol{f}_{\mathrm{obcf}}$ in eq. (25). This can be avoided by design of observers with linear error dynamics that were introduced in [28, 29]. The transformation aims at isolating the nonlinear parts in such a way that nonlinearities only depend on output values. The conditions for their existence are strict, see [19, 30]. Evaluating the integrability conditions of the system given in eq. (18) it can be shown, that there exists no general diffeomorphism, mapping the system eq. (18) to the required form

$$\begin{bmatrix} \dot{\boldsymbol{z}}_{\mathrm{ocf},1} \\ \vdots \\ \dot{\boldsymbol{z}}_{\mathrm{ocf},q} \end{bmatrix} = \begin{bmatrix} \boldsymbol{E}_1 & & \\ & \ddots & \\ & & \boldsymbol{E}_q \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_{\mathrm{ocf},1} \\ \vdots \\ \boldsymbol{z}_{\mathrm{ocf},q} \end{bmatrix} + \begin{bmatrix} \boldsymbol{a}_{\mathrm{ocf},1}(\boldsymbol{y}_{\mathrm{ocf}}, \boldsymbol{u}) \\ \vdots \\ \boldsymbol{a}_{\mathrm{ocf},q}(\boldsymbol{y}_{\mathrm{ocf}}, \boldsymbol{u}) \end{bmatrix}, \quad \boldsymbol{E}_i = \begin{bmatrix} 0 & \boldsymbol{0}^{\mathrm{T}} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \in \mathbb{R}^{p_i \times p_i} \tag{27a}$$

$$\boldsymbol{y}_{\mathrm{ocf}} = \left[h_{\mathrm{ocf},1}(z_{\mathrm{ocf},1,p_1}), \ldots, h_{\mathrm{ocf},q}(z_{\mathrm{ocf},1,p_1}, \ldots, z_{\mathrm{ocf},1,p_q})\right]^{\mathrm{T}}. \tag{27b}$$

However, an observer with linear error dynamics can be designed by taking up the ideas presented in [31, 32, 33]. The main difference to the former approach is a more generalized output map. Reviewing the representation in transformed space eq. (27) reveals an output map depending on selected state variables. Permitting all states in the output map, a diffeomorphism can be derived that encapsulates the nonlinear parts to functions of the measurable outputs. The transformation here is expressed by

$$\boldsymbol{t}_{\text{oled}} = \boldsymbol{z} = \boldsymbol{P}_1(\boldsymbol{x}_1)\boldsymbol{x}_1 + \boldsymbol{P}_2\boldsymbol{J}^{-\text{T}}\boldsymbol{M}\,\text{diag}^{-1}(\boldsymbol{\beta}_e)\boldsymbol{x}_2 + \boldsymbol{P}_3\boldsymbol{x}_3. \tag{28}$$

The matrices $\boldsymbol{P}_i \in \mathbb{R}^{n_s \times n}, i \in \{1,2,3\}$ are design parameters that can be used to trim the characteristics of the observer such as error dynamics and weighting of the measured values. The choice of matrices is restricted by the fact that eq. (28) has to be a diffeomorphism. Since system dependent parameters are included in the transformation, the maximum valid domain of $\boldsymbol{t}_{\text{oled}}$ is specified by

$$\mathcal{D}_{\text{oled}} = \{\boldsymbol{x} \in \mathcal{D}_x : \boldsymbol{v} = \boldsymbol{J}\boldsymbol{x}_2, v_i \neq 0\}. \tag{29}$$

However, this is no further restriction to the observer problem. The observability condition eq. (23) specifies the maximum observable domain independent of the chosen realization. The domain is literally the same as described in eq. (29). Hence, the chosen approach does not decrease the observable domain. Applying transformation eq. (28) to the system yields

$$\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{w}(\boldsymbol{y}, \boldsymbol{u}), \quad \boldsymbol{y} = \boldsymbol{h}_{\text{oled}}(\boldsymbol{z}), \tag{30}$$

with a constant matrix $\boldsymbol{A}$ and output injection $\boldsymbol{w}$. An observer with linear error dynamics can be designed in the new coordinates, if $\boldsymbol{A}$ is Hurwitz.

The derived observer is applied to a planar robot dynamic with two degrees of freedom, see Fig. 8. A friction torque is induced at the two passive joints in the kinematic chains, whereas the passive joint at the end effector is not affected. According to eq. (18) the observed friction parameters are denoted as $\boldsymbol{\tau}_f = [\tau_{f,1}, \tau_{f,2}]$. The dimension of the vector is $n_f = 2 = n$. As shown in section 4.2.1 this is also the maximum number of parameters for this structure.

The friction is observed on the trajectory illustrated in Fig. 8. To avoid the violation of the domain specified in eq. (29) the estimation is paused at very low levels of the passive joints' angular velocity. The output of the observer is presented in Fig. 8. The observer is initialized with zero friction. The estimated output converges to the actual values in short time. Within the interval $t \in [0.1, 0.15]$s the friction is varied linearly. Due to the underlying model a small estimation error remains during this time. After the process the error approaches zero again. The estimated values can be used as a feed forward signal to the controller or to trim the dynamics of the trajectory generator in order to increase the performance of the controller. They can also act as set point variable for a control scheme of the adaptronic joints.

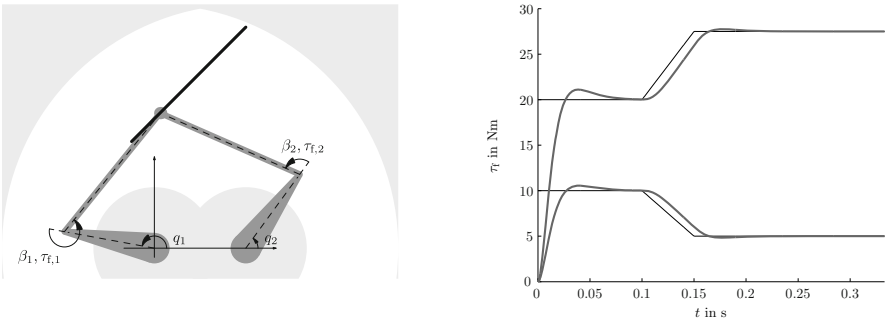


Fig. 8. Left: Sketch of the FIVEBAR-robot and driven path in workspace. Right: Simulative results of friction detection

5 Conclusion

The proposed integrated control scheme matches the general requirements created by the governing manipulation primitive concept. Special properties of parallel kinematic machines are considered and taken into account. Control on drive level is used to encapsulate the robot dynamics and is developed for the purpose of high speed operation. Structural and parametric design has been paid special attention: Computed torque feed forward offers best performance for handling and assembly when parametric design is optimized for disturbance rejection along with the use of a cascaded structure, for experimental results see [3]. Experimental results reveal the performance of integrated motion and force control in a hierarchical framework. Robustness is a keypoint in hybrid control as friction, uncertainties of contact surfaces and of environmental system dynamics parameters and moreover conflicting objectives always have to be considered in handling and assembly. Nonlinear state observers are embedded to estimate friction within passive joints. As uncertainties are eliminated the corrective action of controllers is reduced and at the same time the performance of closed loop control is increased. Furthermore it provides a possibility of including passive joints with variable friction.

References

1. Hasegawa, T., Suehiro, T., Takase, K.: A model-based manipulation system with skill-based execution. IEEE Trans. Robot. Autom. 8(5), 535–544 (1992)
2. Finkemeyer, B.: Robotersteuerungsarchitektur auf der Basis von Aktionsprimitiven. Ph.D. thesis, Technische Universität Braunschweig (2004) (in German)
3. Reisinger, T., Wobbe, F., Kolbus, M., Schumacher, W.: Integrated Force and Motion Control of Parallel Robots – Constrained Space. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 253–273. Springer, Heidelberg (2010)
4. Pavlović, N., Keimer, R., Franke, H.-J.: Adaptronic revolute joints for parallel robots based on quasi-statical clearance adjustment. In: Proceedings of the 17th CISM-IFToMM Symposium, Tokio, Japan, pp. 459–466 (2008)

5. Budde, C.: Wechsel der Konfiguration zur Arbeitsraumvergröerung bei Parallelrobotern. Ph.D. thesis, Technische Universität Braunschweig (2009) (in German)
6. Wobbe, F., Kolbus, M., Schumacher, W.: Enhanced Motion Control Concepts on Parallel Robots. In: Automation and Robotics, pp. 17–40. Tech Education and Publishing (2008) ISBN 9783902613417
7. Stachera, K., Wobbe, F., Schumacher, W.: Jacobian-based derivation of dynamics equations of elastic parallel manipulators. In: Proc. of the IASTED Asian Conference on Modelling and Simulation (AsiaMS 2007), Beijing, China (2007)
8. Sciavicco, L., Siciliano, B.: Modelling and control of robot manipulators. Springer, Berlin (2005) ISBN 1852332212
9. Föllinger, O.: Nichtlineare Regelungen, Bd.2, Harmonische Balance, Popowkriterium und Kreiskriterium, Hyperstabilität, Synthese im Zustandsraum. Oldenbourg, 7^{th} edition (1993) ISBN 9783486225037
10. Brunotte, C.: Regelung und Identifizierung von Linearmotoren für Werkzeugmaschinen. Ph.D. thesis, TU Braunschweig (1999)
11. Brandenburg, G., Papiernik, W.: Feedforward and feedback strategies applying the principle of input balancing for minimal tracking errors in cnc machine tools. In: Proc. 4th International Workshop on Advanced Motion Control, AMC 1996-MIE, vol. 2, pp. 612–618 (1996)
12. Wobbe, F., Böske, W., Schumacher, W.: Optimierte Antriebsreglerstrukturen zur Störunterdrückung. In: SPS/IPC/DRIVES 2006, Nürnberg, Germany, pp. 473–482 (2006)
13. Utkin, V.I.: A survey: Variable structure systems with sliding modes. IEEE Trans. Autom. Control 22(2), 212–222 (1977)
14. Slotine, J.-J.: Tracking Control of Nonlinear Systems Using Sliding Surfaces. Ph.D. thesis, Massachusetts Institute of Technology (1983)
15. Filippov, A.F.: Differential Equations with Discontinuous Righthand Sides: Control Systems (Mathematics and Its Applications), 1st edn. Springer, Netherlands (1988) ISBN 9789027726995
16. Wobbe, F., Kolbus, M., Schumacher, W.: Continuous sliding surfaces versus classical control concepts on parallel robots. In: Proc. of the 13th IEEE IFAC International Conference on Methods and Models in Automation, Szczecin, Poland (2007)
17. Levant, A., Friedman, L.: Higher Order Sliding Modes. In: Levant, A., Friedman, L. (eds.) Sliding Mode Control in Engineering Control Engineering Series, 11, pp. 53–101. Marcel Dekker, Inc., New York (2002)
18. Britton, N.F., Garces, F.R., Becerra, V.M.: Strategies for Feedback Linearisation: A Dynamic Neural Network Approach (Advances in Industrial Control), 1st edn. Springer, Berlin (2003) ISBN 9781852335014
19. Isidori, A.: Nonlinear Control Systems 1. An Introduction (Practical Astronomy), 3rd edn. Springer, Berlin (1995) ISBN 9783540199168
20. Whitcomb, L.L., Rizzi, A.A., Koditschek, D.E.: Comparative experiments with a new adaptive controller for robot arms. IEEE Trans. Robot. Autom. 9(1), 59–70 (1993)
21. Slotine, J.-J.: The robust control of robot manipulators. International Journal of Robotics Research 4(2), 49–64 (1985)
22. Budde, C., Last, P., Hesselbach, J.: Workspace enlargement of a triglide robot by changing working assembly mode. In: Proceedings of the IASTED Internatioanl Conference Robotics and Applications, RA, Cambridge, MA, USA, pp. 244–248 (2005)
23. Pavlović, N., Keimer, R.: Improvement of overall performance of parallel robots by adapting friction of joints using quasi-statical clearance adjustment. In: Proceedings of Adaptronic Congress, Berlin, pp. 179–183 (2008)

24. Maaß, J., Kolbus, M., Wobbe, F., Schumacher, W., Bier, C., Raatz, A., Hesselbach, J.: Advances in motion control for high-performance parallel robots. In: Proc. of the 3rd International Colloquium of the Collaborative Research Center SFB, vol. 562, pp. 101–112. Shaker Verlag, Braunschweig (2008)
25. Birk, J.: Rechnergestützte Analyse und Lösung nichtlinearer Beobachtungsaufgaben. Ph.D. thesis, Universität Stuttgart (1992) (in German)
26. Nicosia, S., Tormnambe, A.: High-gain observers in the state and parameter estimation of robots having elastic joints. Systems & Control Letters 12, 47–52 (1989) ISSN 331-337
27. Besançon, G. (ed.): Nonlinear Observers and Applications, 1st edn. Lecture Notes in Control and Information Sciences. Springer, Berlin (2007)
28. Bestle, D., Zeitz, M.: Canonical form observer design for non-linear time-variable systems. International Journal of Control 38, 419–431 (1983)
29. Krener, A.J., Isidori, A.: Linearization by output injection and nonlinear observers. Systems & Control Letters 3(1), 47–52 (1983)
30. Keller, H.: Entwurf nichtlinearer Beobachter mittels Normalformen. Ph.D. thesis, Universität Karlsruhe (1986) (in German)
31. Kazantzis, N., Kravaris, C.: Nonlinear observer design using lyapunov's auxiliary theorem. In: Proceedings of the 36th IEEE Conference on Decision and Control, vol. 5, pp. 4802–4807 (1997)
32. Krener, A.J., Xiao, M.: Nonlinear observer design in the siegel domain. SIAM Journal on Control and Optimization 41(3), 932–953 (2002)
33. Haddadin, S., Albu-Schaffer, A., De Luca, A., Hirzinger, G.: Collision detection and reaction: A contribution to safe physical human-robot interaction. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008, pp. 3356–3363 (2008)

Integrated Force and Motion Control of Parallel Robots – Part 2: Constrained Space

Thomas Reisinger, Frank Wobbe, Michael Kolbus, and Walter Schumacher

Abstract. Parallel robots in the context of handling and assembly benefit from their inherent high dynamics. However, their coupled nonlinear system dynamics demand for an effective control system in terms of accuracy, robustness and disturbance rejection. This paper presents a hybrid control scheme that is designed specifically to these features. The hierarchical approach encapsulates the robots dynamics on drive-level and, thus, provides the basis for effective real-time hybrid control. It is interfaced by a fully specified C^2-continuous trajectory, see [1] for detailed design. Force-torque control is derived and an analysis of controller performance during the transition to constrained motion presented. These modules form a unique, integrated control-scheme.

1 Introduction

The control concept introduced in this paper focuses on force and motion control of parallel robots in constrained space. Due to task definitions in the field of handling and assembly it is optimized with respect to high dynamics. It bases on the subordinated drive control with encapsulated dynamics and is embedded in the software infrastructure developed at the collaborative research center SFB 562. This infrastructure provides a complete functional chain from task generation to its execution on physical layer. Additional demands for the control concepts are derived from the fact that tasks are specified by Manipulation Primitives, see [2].

Thomas Reisinger
ABB Corporate Research Center, Wallstadter Straße 59, 68526 Ladenburg, Germany
e-mail: thomas.reisinger@de.abb.com

Frank Wobbe · Michael Kolbus · Walter Schumacher
Technische Universität Braunschweig, Institute of Control Engineering,
Hans-Sommer-Straße 66, 38106 Braunschweig, Germany
e-mail: {wobbe,kolbus,w.schumacher}@ifr.ing.tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 253–273.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

As presented in [1] the overall framework of the control architecture yields a highly modular design with encapsulated dynamics. In addition to a drive level controller with a specified interface, a unit is presented that adds the functionality of force control. In combination with position and velocity units, hybrid control in operational space is provided.

The paper is organized as follows. Based on the characterization of interaction dynamics, the design of force control is presented in section 2. Design aspects are discussed in detail. Experiments and results are presented and discussed in the following section 3. The last section summarizes the results and draws a conclusion.

2 Force Control

The force control concept for parallel robots based on Manipulation Primitives aims to describe a holistic approach. All effects occurring in the process where the end effector establishes contact with the environment are taken into account.

The constraint system dynamics introduces an environment dependent interaction frame. Thus, forces and torques in the task frame are modeled during a robot task. Impacts of the robot's end effector on the environment surface and the system behavior during the transition phase are described using non-smooth mechanics, namely *Moreau's sweeping* process.

2.1 Force Control in Assembly

Interaction with the environment by force control can be classified into two groups. Schemes using a direct force feedback loop are referred to as direct force control [3], whereas the ones using motion control with force feedback as additional constraint reference group the indirect force control [4], such as impedance or compliance control. In latter ones position errors are related to contact forces via stiffness analogous to a mass-damper-system. This guarantees robustness in terms of uncertain environment boundaries, but lacks an a-priori precisely known contact-force at equilibrium-point. Direct force control meets this drawback and enables hybrid force control along the constraint and motion control in the direction of the unconstrained task directions. An overview can be found in [5]. In the context of parallel robots the complexity of the DKP is a major disadvantage. Hence, it narrows the use of purely operational space based control concepts. An alternative approach is presented in the forthcoming sections. Moreover, special attention has to be paid to complementary conditions. Force control presumes an established contact, whereas motion control requires an unconstrained environment. In handling and assembly however, task specifications blur the boundaries as they demand for both. The complementary problem formulation for each part itself is easy to be met by a control scheme – the transition phase in between, however, cannot be neglected and as pointed out in [6] is mandatory for closed stability and performance, as contact establishment involves a sweeping process, cf. [7] and [8].

2.2 Interaction with the Environment

If an end effector of a parallel robot or a firmly grasped object touches the environment surface, external forces and torques occur and influence the system behavior. The direction as well as the value of these generalized external forces need to be determined in order to develop an environment model for the force control. This problem first was encountered in [9] and solved for holonomic forces exerted by static environments.

A comprehensive investigation on the phenomena during a typical force control task shows three phases (cf. [10]). During the approach of the end effector toward the surface, the robots moves in the unconstrained workspace and the description used for the *subordinate drive controller* [1] remains valid. The transition phase, where the end effector establishes contact and non-smooth behavior takes place, is described by the transition model. In contrast, the contact model describes the dynamics while the end effector is in continuous contact with the surface. Together they form the environment model derived in the following sections.

2.2.1 Constraints Formulation

Workpieces, loading equipment, conveyors and peripheral devices in a robot cell are necessary to accomplish an assembly task. Together all these objects define the *environment* and impose constraints on the robot's motion leading to external forces that need to be controlled. The geometry of the environment determines the direction of these external forces. According to [10], the environment is an union set of m smooth surfaces $\Gamma(\boldsymbol{r})$ in three-dimensional space. Hence, in the subspace

$$\Phi = \bigcap_{i=1}^{m} \{\boldsymbol{r} : \Gamma_i(\boldsymbol{r}) \geq 0\} \tag{1}$$

only constrained motions of the parallel robot are feasible. It is assumed, that the contact is *unilateral*. Thus, the end effector or the grasped workpiece is in contact with the environment at one point only. Considering a rigid environment, the desired path of the contact point $\boldsymbol{r}_{\mathrm{des},c}(t)$ is deflected on the surface and modified into the actual contact point $\boldsymbol{r}_c(t)$. Resulting end effector forces are aligned with the outward pointing normal vector of the environment surface at the actual contact point, determined by

$$\boldsymbol{n}_c(t) = \frac{\nabla\Gamma(\boldsymbol{r}_c(t))}{\|\nabla\Gamma(\boldsymbol{r}_c(t))\|_2} \tag{2a}$$

while the tangent vector

$$\tilde{\boldsymbol{t}}_c(t) = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{r}_c(t) \tag{2b}$$

at the same position shows the direction of the motion. The normal vector $\boldsymbol{n}_c(t)$ and normalized tangent vector $\boldsymbol{t}_c(t)$ together with the yielding binormal vector $\boldsymbol{b}_c(t)$ define the interaction frame and the corresponding homogeneous transformation in the robot's base frame:

$$ {}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{IF}}(t) = \begin{bmatrix} \boldsymbol{t}_c(t) & \boldsymbol{b}_c(t) & \boldsymbol{n}_c(t) & {}^{\mathrm{BF}}\boldsymbol{r}_c(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^{\mathrm{BF}}\boldsymbol{R}_c(t) & {}^{\mathrm{BF}}\boldsymbol{r}_c(t) \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix}. \tag{3} $$

Fig. 1 shows the deflection of a desired end effector path by the environment surface and the resulting direction of the forces in the interaction frame. The parallel robot is located inside the sphere and the desired path $\boldsymbol{r}_c(t)$ connects point t'_a directly with t'_b. Its projection on the x-y-plane is displayed as the dashed line. Due to the geometry of the environment, the actual path follows the sphere as indicated by the solid line from t'_a to t'_b. At time t', the end effector forces point outward along $\boldsymbol{n}_c(t')$ and $\boldsymbol{t}_c(t')$ shows the direction of the motion.

The position and force control takes place in the task frame, while the induced forces are determined by the interaction frame (3). Hence, the homogeneous transformation

$$ {}^{\mathrm{TF}}\boldsymbol{T}_{\mathrm{IF}}(t) = {}^{\mathrm{BF}}\boldsymbol{T}^{-1}_{\mathrm{TF}}(t)\,{}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{IF}}(t) = \begin{bmatrix} {}^{\mathrm{BF}}\boldsymbol{R}^{\mathrm{T}}_{\mathrm{TF}}\,{}^{\mathrm{BF}}\boldsymbol{R}_{\mathrm{IF}} & {}^{\mathrm{BF}}\boldsymbol{R}^{\mathrm{T}}_{\mathrm{TF}}\left({}^{\mathrm{BF}}\boldsymbol{r}_c - {}^{\mathrm{BF}}\boldsymbol{r}_{\mathrm{TF}}\right) \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix} \tag{4} $$

determines the forces in the task frame which are

$$ {}^{\mathrm{TF}}\boldsymbol{f} = \begin{bmatrix} {}^{\mathrm{BF}}\boldsymbol{R}^{\mathrm{T}}_{\mathrm{TF}}\,{}^{\mathrm{BF}}\boldsymbol{R}_{\mathrm{IF}}\,{}^{\mathrm{IF}}\boldsymbol{f} \\ \left({}^{\mathrm{BF}}\boldsymbol{R}^{\mathrm{T}}_{\mathrm{TF}}\left({}^{\mathrm{BF}}\boldsymbol{r}_c - {}^{\mathrm{BF}}\boldsymbol{r}_{\mathrm{TF}}\right)\right) \times \left({}^{\mathrm{BF}}\boldsymbol{R}^{\mathrm{T}}_{\mathrm{TF}}\,{}^{\mathrm{BF}}\boldsymbol{R}_{\mathrm{IF}}\,{}^{\mathrm{IF}}\boldsymbol{f}\right) \end{bmatrix}. \tag{5} $$

Above calculations assume a fixed task frame during the whole force controlled task. Analogous calculations to determine the forces for the various cases of a moving task frame behavior and the robot's base frame exist and are described in [6].

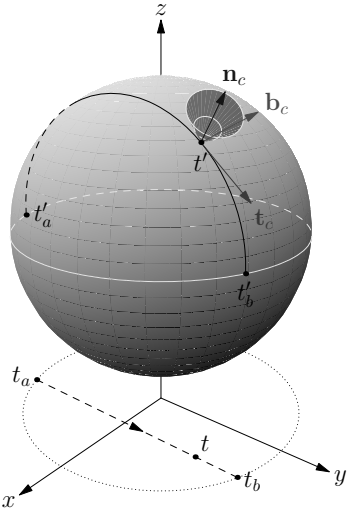


Fig. 1. Constrained path of the end effector

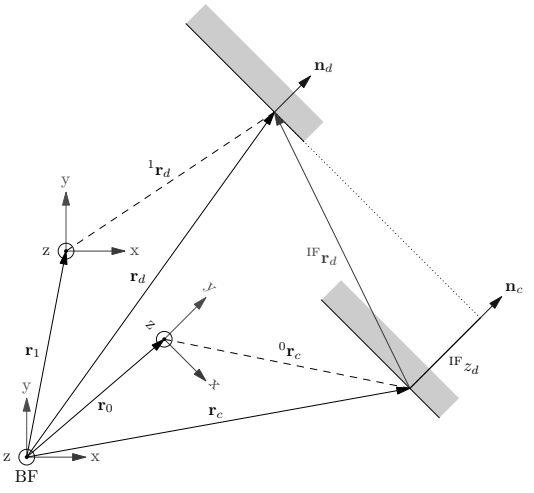


Fig. 2. Environment displacement in the robot's base frame

2.2.2 Modeling the Environment

Using the constraints formulation, the environment and the direction of the external forces can be described. To be able to solve the force control problem for parallel robots, the value of the external forces need to be determined. By taking a holistic approach for the force control scheme, the contact phase as well as the transition phase during a force control task need to be taken into account.

Contact Model

During the contact phase, the end effector or an attached workpiece is settled on the environment surface. The exerted forces are determined by the dynamics of the robot and the environment as well as by the implemented control algorithm of the robot.

The surface is deflected by moving the robot's end effector while in contact with the environment according to the situation shown in Fig. 2. At the beginning of the force control task, the end effector pose is described by the transformation ${}^{\mathrm{BF}}\boldsymbol{T}_0$. After motion control in unconstrained space, pose ${}^{\mathrm{BF}}\boldsymbol{T}_1$ is reached and the surface is deflected by ${}^{\mathrm{IF}}z_d$ along the surface's normal $\boldsymbol{n}_c$. It is assumed that the deflection is very small and external forces induced by the rotation can be neglected. Hence, the deflection can be calculated by

$$ {}^{\mathrm{IF}}z_d = \boldsymbol{s}^{\mathrm{T}}\boldsymbol{D}^{\mathrm{T}}\,{}^{\mathrm{IF}}\boldsymbol{T}_d({}^{\mathrm{BF}}\mathrm{d}\boldsymbol{x})\boldsymbol{t}, \tag{6} $$

where the operators $\boldsymbol{D}$, $\boldsymbol{s}$ and $\boldsymbol{t}$ are defined by the task frame formalism in [11] and ${}^{\mathrm{BF}}\mathrm{d}\boldsymbol{x}$ is the described end effector motion in the robot's base frame. The deflection is the input for a one-dimensional environmental model resembling a simple stiffness

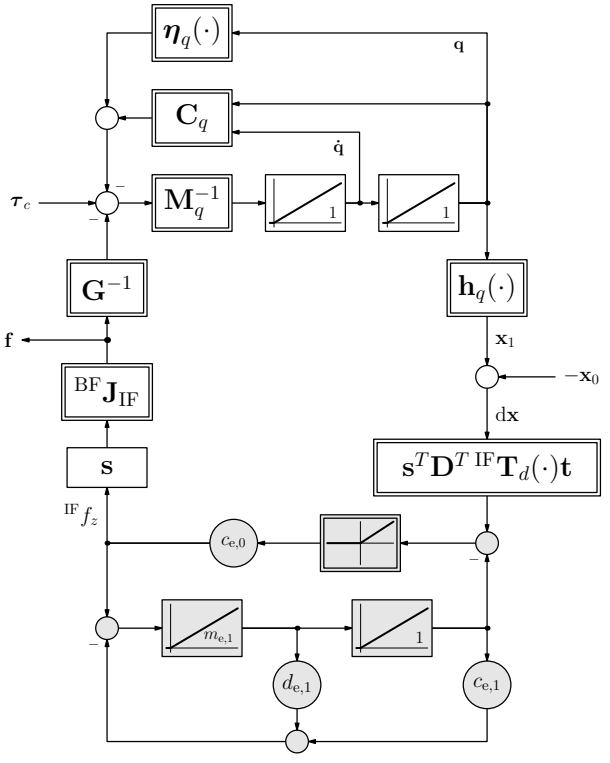


Fig. 3. Parallel robot in contact with environment

or a more complex model including spring and damping parameters and generating an external force

$$ {}^{\mathrm{IF}}f_z = g({}^{\mathrm{IF}}z_d). \tag{7} $$

Finally, this force is transformed with the Jacobian ${}^{\mathrm{BF}}\boldsymbol{J}_{\mathrm{IF}}$ from the interaction into the base frame according to

$$ {}^{\mathrm{BF}}\boldsymbol{f} = {}^{\mathrm{BF}}\boldsymbol{J}_{\mathrm{IF}}({}^{\mathrm{BF}}\boldsymbol{x}_d)\boldsymbol{s}\,{}^{\mathrm{IF}}f_z = -\boldsymbol{f}_{\mathrm{ext}} \tag{8} $$

and the dynamics model for constrained robot motions

$$ \boldsymbol{M}_q(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_q(\dot{\boldsymbol{q}}, \boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{\eta}_q(\boldsymbol{q}) + \boldsymbol{G}^{-1\,\mathrm{BF}}\boldsymbol{f} = \boldsymbol{\tau} \tag{9} $$

is achieved. Here, ${}^{\mathrm{BF}}\boldsymbol{f}$ denotes the torques and forces generated by the parallel robot to compensate external forces. The block diagram in Fig. 3 summarizes the calculations above. All transformations are denoted in nonlinear blocks and the dynamic contact model uses gray-shaded boxes to distinguish it from the rest of the system.

Transition Model

Due to uncertainties in path planning, tolerances of the environment and high velocities during the approach, the end effector hits the surface and may bounce back into unconstrained space. This may happen several times during the transition phase while the end effector establishes contact with the environment. The system behavior is different from the differential equation derived above but can be described as a non-smooth mechanical problem using *Moreau's sweeping process* (cf. [7]).

When the end effector hits a surface, the general dynamics equation of an Euler-Lagrange-system

$$\boldsymbol{M}(\boldsymbol{x},t)\ddot{\boldsymbol{x}} - \boldsymbol{p}(\boldsymbol{x},\dot{\boldsymbol{x}},t) = \boldsymbol{r}(t), \tag{10}$$

remains no longer valid. Here, $-\boldsymbol{p}$ is the vector field of the Coriolis as well as the gravitational forces and $\boldsymbol{r}$ defines a general reaction as a system input. Instead, the velocity $\dot{\boldsymbol{x}}$ is discontinuous at time t_k where the impact occurs and is replaced by a *velocity of bounded variation*. The latter one is defined via its derivative

$$\mathrm{d}\boldsymbol{u}(\{t_k\}) = \int_{\{t_k\}} \mathrm{d}\boldsymbol{u} = \boldsymbol{u}(t_k^+) - \boldsymbol{u}(t_k^-), \tag{11}$$

which is a measure and therefore also is defined at impact time instant t_k. Concurrently, the measure of $\{t_k\}$ is zero. Hence, the term of the vector field reduces to

$$\int_{\{t_k\}} \boldsymbol{p}(\boldsymbol{x},\dot{\boldsymbol{x}},t)\mathrm{d}t = 0 \tag{12}$$

leading to the significant finding

$$\mathrm{d}\boldsymbol{r}(\{t_k\}) = \boldsymbol{M}(\boldsymbol{x},t_k)\mathrm{d}\boldsymbol{u}(\{t_k\}) = \boldsymbol{M}(\boldsymbol{x},t_k)\Big[\boldsymbol{u}(t_k^+) - \boldsymbol{u}(t_k^-)\Big]. \tag{13}$$

The above equation shows that the system reaction $\mathrm{d}\boldsymbol{r}$ depends on the system inertia $\boldsymbol{M}$ and the velocity difference before $\boldsymbol{u}(t_k^-)$ and after $\boldsymbol{u}(t_k^+)$ the impact, only. This has to be taken into account during synthesis of the transition controller.

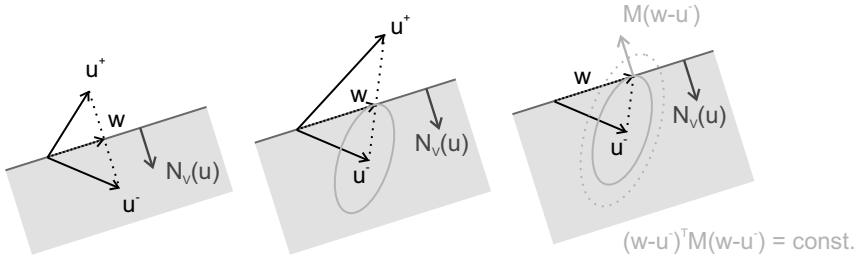
According to *Moreau's sweeping process*, the system reaction $-\mathrm{d}\boldsymbol{r}$ at the instant of impact is an element of the outward normal cone $N_{V(\boldsymbol{x}(t))}(\boldsymbol{u}(t))$, defined by the environment surface and the pre- and post-impact velocities. Hence, the system behavior in the transition phase can be described as

$$-\mathrm{d}\boldsymbol{r} = \boldsymbol{p}(\boldsymbol{x},\boldsymbol{u},t)\mathrm{d}t - \boldsymbol{M}(\boldsymbol{x},t)\mathrm{d}\boldsymbol{u} \in N_{V(\boldsymbol{x}(t))}(\boldsymbol{u}(t)). \tag{14}$$

Due to a discretization of the above equation, the robots behavior during the non-smooth transition phase can be simulated and thus, the implemented control algorithms can be verified. Using generalized velocity in the tangent space

$$\boldsymbol{w} := \frac{1}{1+e}\boldsymbol{u}(t_k^+) + \frac{e}{1+e}\boldsymbol{u}(t_k^-), \tag{15}$$

with e as Newtons coefficient of restitution, eq. (13) can be reformulated



(a) Impact for point-masses (b) Generalized impact (c) Projection considering mass-distribution

Fig. 4. Impact and reaction

$$\begin{aligned} -N_{V(\boldsymbol{x}(t_k))}(\boldsymbol{u}(t_k)) \ni \mathrm{d}\boldsymbol{r}(\{t_k\}) &= \boldsymbol{M}(\boldsymbol{x},t_k)\left[\boldsymbol{u}(t_k^+) - \boldsymbol{u}(t_k^-)\right] \\ &= \kappa \boldsymbol{M}(\boldsymbol{x},t_k)\left[\boldsymbol{w} - \boldsymbol{u}(t_k^-)\right], \quad \kappa = (1+e) \in \mathbb{R}^+. \end{aligned} \tag{16}$$

Here $\boldsymbol{w} \in V(\boldsymbol{x}(t_k))$ is a solution of an elliptic minimization-problem

$$\frac{1}{2}\left(\boldsymbol{w} - \boldsymbol{u}(t_k^-)\right)^{\mathrm{T}} \boldsymbol{M}(\boldsymbol{x},t_k)\left(\boldsymbol{w} - \boldsymbol{u}(t_k^-)\right) = \text{const}$$

to the subspace V, as $\boldsymbol{M}(\boldsymbol{x},t_k)\left(\boldsymbol{w} - \boldsymbol{u}(t_k^-)\right) \in -N_{V(\boldsymbol{x}(t_k))}(\boldsymbol{u}(t_k))$, cf. Fig. 4. Considering this as an variational inequality and using projection leads with eq. (15) to

$$\boldsymbol{u}(t^+) = -e\boldsymbol{u}(t^-) + (1+e)\,\mathrm{prox}_{\boldsymbol{M}(\boldsymbol{x},t)}\left(V(\boldsymbol{x}(t)); \boldsymbol{u}(t^-)\right) \tag{17}$$

as analytical solution. The equation refers to the point in time when the impact occurs, and thus the index k for a generalized time-step has been skipped. Post- and pre-impact velocities are indicated by $\boldsymbol{u}(t^+)$ and $\boldsymbol{u}(t^-)$, respectively. Here, the operator $\mathrm{prox}_{\boldsymbol{A}}(T; \boldsymbol{v}) := \mathrm{argmin}_{\boldsymbol{x}\in T} \frac{1}{2}(\boldsymbol{x}-\boldsymbol{v})^{\mathrm{T}} \boldsymbol{A}(\boldsymbol{x}-\boldsymbol{v})$ denotes the projection of $\boldsymbol{v}$ onto the closed convex nonempty set T, w.r.t. the metric defined by $\boldsymbol{A}$. Introducing $\boldsymbol{\nu} = \boldsymbol{u}(t^-) - \boldsymbol{w}$ as difference vector gives

$$\begin{aligned} -\boldsymbol{w} = \boldsymbol{u}(t^-) - \boldsymbol{w} - \boldsymbol{u}(t^-) &= \boldsymbol{\nu} - \boldsymbol{u}(t^-) \\ &= \boldsymbol{M}^{-1}(\boldsymbol{x},t)\left(\boldsymbol{M}(\boldsymbol{x},t)\boldsymbol{\nu} - \boldsymbol{M}(\boldsymbol{x},t)\boldsymbol{u}(t^-)\right) \quad \in -V(\boldsymbol{x}(t)). \end{aligned} \tag{18}$$

Considering eq. (18) and the fact that $\boldsymbol{M}(\boldsymbol{x},t)\boldsymbol{\nu} \in N_{V(\boldsymbol{x}(t))}(\boldsymbol{u}(t))$ is valid, $\boldsymbol{M}(\boldsymbol{x},t)\boldsymbol{\nu}$ is the solution of an elliptic minimization problem

$$\frac{1}{2}\left(\boldsymbol{M}(\boldsymbol{x},t)\boldsymbol{\nu} - \boldsymbol{M}(\boldsymbol{x},t)\boldsymbol{u}(t^-)\right)^{\mathrm{T}} \boldsymbol{M}^{-1}(\boldsymbol{x},t)\left(\boldsymbol{M}(\boldsymbol{x},t)\boldsymbol{\nu} - \boldsymbol{M}(\boldsymbol{x},t)\boldsymbol{u}(t^-)\right) = \text{const}$$

to the subspace $N_V(\boldsymbol{u})$. Using projection and left-hand side multiplication by inverse matrix $\boldsymbol{M}^{-1}(\boldsymbol{x},t)$ thus results in projected velocity $\boldsymbol{\nu} \in N_V(\boldsymbol{u})$. Substituting the solutions for velocities $\boldsymbol{w}$ and $\boldsymbol{\nu}$ obtained by projection

$$\begin{aligned}\boldsymbol{w} &= \operatorname{prox}_{\boldsymbol{M}(\boldsymbol{x},t)}\left(V(\boldsymbol{x}(t));\, \boldsymbol{u}(t^-)\right)\\ \boldsymbol{\nu} &= \boldsymbol{M}^{-1}(\boldsymbol{x},t)\operatorname{prox}_{\boldsymbol{M}^{-1}(\boldsymbol{x},t)}\left(N_{V(\boldsymbol{x}(t))}\left(\frac{1}{1+e}\boldsymbol{u}(t^+)+\frac{e}{1+e}\boldsymbol{u}(t^-)\right);\, \boldsymbol{M}(\boldsymbol{x},t)\boldsymbol{u}(t^-)\right)\end{aligned} \tag{19}$$

into

$$\boldsymbol{w} = \boldsymbol{u}(t^-) - \boldsymbol{\nu} \tag{20}$$

eq. (17) can be reformulated to

$$\begin{aligned}&\boldsymbol{u}(t^+) = \boldsymbol{u}(t^-)\\ &-(1+e)\boldsymbol{M}^{-1}(\boldsymbol{x},t)\operatorname{prox}_{\boldsymbol{M}^{-1}(\boldsymbol{x},t)}\left(N_{V(\boldsymbol{x}(t))}\left(\frac{1}{1+e}\boldsymbol{u}(t^+)+\frac{e}{1+e}\boldsymbol{u}(t^-)\right);\, \boldsymbol{M}(\boldsymbol{x},t)\boldsymbol{u}(t^-)\right).\end{aligned} \tag{21}$$

In case of a smooth surface the outer normal cone reduces to a vector and thus the proximum resolves to $\operatorname{prox}_{\boldsymbol{A}}(T;\, \boldsymbol{v}) = \left[\boldsymbol{t}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{t}\right]^{-1}\boldsymbol{t}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}\boldsymbol{t}$ with $\boldsymbol{t} \in T$.

For unilateral contacts and smooth surfaces thus eq. (21) turns into

$$\begin{aligned}&\boldsymbol{u}(t^+) = \boldsymbol{u}(t^-)\\ &-(1+e)\boldsymbol{M}^{-1}(\boldsymbol{x},t)\nabla\Gamma_i(\boldsymbol{x}(t))\left[\nabla\Gamma_i(\boldsymbol{x}(t))^{\mathrm{T}}\boldsymbol{M}^{-1}(\boldsymbol{x},t)\nabla\Gamma_i(\boldsymbol{x}(t))\right]^{-1}\nabla\Gamma_i(\boldsymbol{x}(t))^{\mathrm{T}}\boldsymbol{u}(t^-).\end{aligned} \tag{22}$$

Hence (17) and (21) describe simple frictionless sweeping processes that can be solved analytically [8]. It is important that the post-impact velocity $\boldsymbol{u}(t^+)$ depends on four factors: pre-impact velocity $\boldsymbol{u}(t^-)$, pose-dependent mass-matrix $\boldsymbol{M}(\boldsymbol{x},t)$, the normal-vector $\nabla\Gamma_i(\boldsymbol{x}(t))$ of the surface and surface dynamics, here taken into account by Newtons coefficient of restitution e. These variable factors render the post-impact uncertain. Therefore, force control has to cope with these uncertainties, as the transition phase cannot be neglected. However, it can be tackled by introducing a damping phase in control which will be presented in chapter 2.3.2.

2.3 Force Control Based on Subordinated Drive Controller

The force control concept for the contact and transition phase is based on the subordinate drive controller with feed forward linearization (cf. [1]) and is shown in Fig. 5. On the left hand side, there are the elements of the motion module, planning the robot's motion in the task frame according to [11]. While on the right hand side, the subordinate drive control with environment model is depicted. Transfer function $\boldsymbol{G}_{K_3}$ symbolizes the general force control algorithm. Its output, the control variable, is fed into a third order feed forward trajectory generator (PT_3) to provide a fully specified C^2-continuous trajectory for the task frame formalism. The parameters of the trajectory generator are adjusted in such a way that the parallel robot in contact with the environment can follow the reference trajectory in every pose inside

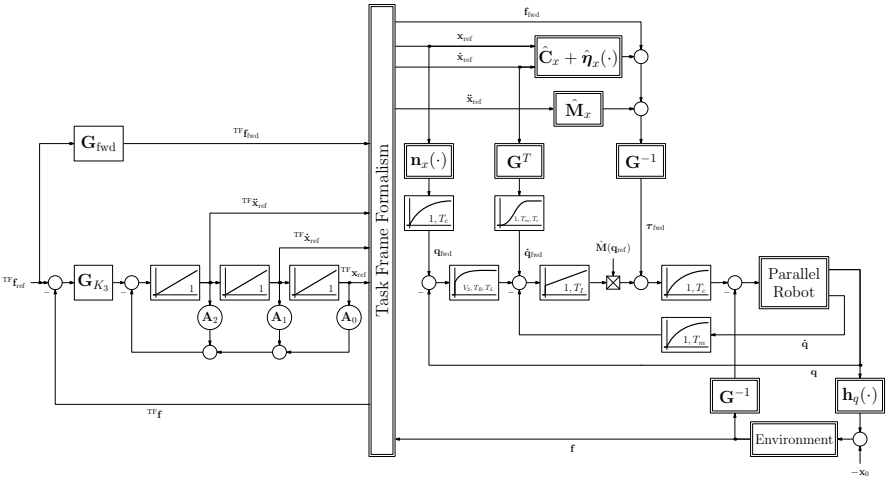


Fig. 5. Block schematic drawing of the overall force control concept based on subordinate drive controller

its workspace. Hence, the trajectory generator encapsulates the position controlled robot and the latter one can be neglected for the synthesis of the control algorithms.

Additionally, a force feed forward signal is provided to improve the dynamic behavior of the force controller.

2.3.1 Contact Phase

Assuming the end effector or the tightly grasped object is in stable contact with the environment, the force controller regulates the interaction of both systems. It is discussed for a static environment, a *PI*-controller in every degree-of-freedom $\langle \mathrm{k} \rangle$ of the task frame satisfies the need for a vanishing control offset and stable system behavior

$$G_{K_3}^{\langle \mathrm{k} \rangle}(s) = V_3^{\langle \mathrm{k} \rangle} \frac{T_{i,3}^{\langle \mathrm{k} \rangle} s + 1}{T_{i,3}^{\langle \mathrm{k} \rangle} s}, \quad k \in [1; n]_{\mathbb{N}}. \tag{23}$$

Taking into account that the trajectory generator encapsulates the robot's constrained system dynamics, the transfer function of the open force control loop consists of the controller $G_{K_3}^{\langle \mathrm{k} \rangle}(s)$ itself, the trajectory generator $G_3^{\langle \mathrm{k} \rangle}(s)$ and the contact model $G_E^{\langle \mathrm{k} \rangle}(s)$. This yields

$$F_{o,3}^{\langle \mathrm{k} \rangle}(s) = \left(G_{K_3}^{\langle \mathrm{k} \rangle} G_3^{\langle \mathrm{k} \rangle} G_E^{\langle \mathrm{k} \rangle} \right)(s) = V_3^{\langle \mathrm{k} \rangle} \frac{T_{i,3}^{\langle \mathrm{k} \rangle} s + 1}{T_{i,3}^{\langle \mathrm{k} \rangle} s} \frac{c_{\mathrm{e}}^{\langle \mathrm{k} \rangle} l^{\langle \mathrm{k} \rangle 2}}{(T_a s + 1)\left(\frac{s^2}{\omega_{0,a}^2} + \frac{2D_a}{\omega_{0,a}} s + 1 \right)} \tag{24}$$

and by using the controller's zero to cancel the trajectory generator's pole

$$F_{o,3}^{\langle \mathrm{k} \rangle} = \frac{V_3^{\langle \mathrm{k} \rangle} \cdot c_\mathrm{e}^{\langle \mathrm{k} \rangle} l^{\langle \mathrm{k} \rangle^2}}{T_a s \left(\frac{s^2}{\omega_{0,a}^2} + \frac{2D_a}{\omega_{0,a}} s + 1 \right)}. \tag{25}$$

By defining a desired phase margin, the remaining controller gain $V_3^{\langle \mathrm{k} \rangle}$ can be calculated. The transfer function for generating the force feed forward signal is of 1st order type (PT_1) for the closed force feedback loop. It is given by defining the equivalent time constant for the closed force feedback loop.

The contact model for a static environment with stiffness $c_\mathrm{e}^{\langle \mathrm{k} \rangle}$ contains a parameter $l^{\langle \mathrm{k} \rangle}$ representing a crank for torque generation. This parameter is relevant for torque control and thus needs to be set up carefully for the rotational degrees of freedom where torque control takes place.

2.3.2 Transition Phase

In contrast to the previously described contact phase, the end effector is not yet settled during transition and impacts may occur. Nevertheless, this is an inevitable behavior in a realistic scenario for sensor based task execution with geometry tolerances and unknown environments. To establish the contact and to start the assembly task, the robot approaches the environment surface with a velocity-limited trajectory. As soon as the first impact is detected by the force sensor, the damping of the transient behavior is activated. A finite state machine assures the right controller selection depending on the system state.

The damping trajectory fed into the task frame formalism (cf. Fig. 5), samples and holds the position of the first impact and sets velocity as well as acceleration as reference for motion control to zero:

$$T_{\mathrm{ref},d}^{\langle \mathrm{k} \rangle} = \left\{ {}^{\mathrm{TF}}x \Big|_{t\,=\,t_k}, 0, 0 \right\} \quad \forall\, t \in [t_k; t_d]. \tag{26}$$

Hence, the subordinate drive controller aims to establish a contact situation and makes sure that the bouncing end effector is always redirected toward the environment surface. Employing *Moreau's sweeping process* ensures the stability of the system as illustrated below.

The dynamics model of the damped system according to the nomenclature defined in section 2.2.2 and [7] is

$$\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} - \boldsymbol{p} = \boldsymbol{r}. \tag{27}$$

Above equation describes the system behavior in Cartesian coordinates with the vector field of Coriolis and centripetal forces defined as

$$\boldsymbol{p}(\boldsymbol{h}(\boldsymbol{q}), \boldsymbol{G}^{-\mathrm{T}}\dot{\boldsymbol{q}}, \boldsymbol{x}_\mathrm{ref}, \dot{\boldsymbol{x}}_\mathrm{ref}, \ddot{\boldsymbol{x}}_\mathrm{ref}). \tag{28}$$

analogous to eq. (10). Equations (27) and (28) describe the robot's dynamic behavior with respect to *position control*, cf. the right part of Fig. 5 where the subordinated

drive controller is additionally taken into account. Reference values are denoted in Cartesian coordinates and actual values in joint coordinates with the respective transformations to emphasize that the control loop simulation of the damping takes place in the subordinate drive controller.

Taking into account the reference trajectory during the damping phase (26) yields the system description according to

$$\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} - \boldsymbol{p}(\boldsymbol{h}(\boldsymbol{q}), \boldsymbol{G}^{-\mathrm{T}}\dot{\boldsymbol{q}}, \boldsymbol{x}_{\mathrm{ref}}, \boldsymbol{0}, \boldsymbol{0}) = \boldsymbol{r}. \tag{29}$$

For simulation purposes numeric inversion of the inertia matrix is required. Finally, the system behavior in the transient phase is

$$\ddot{\boldsymbol{x}} - \tilde{\boldsymbol{p}}(\boldsymbol{h}(\boldsymbol{q}), \boldsymbol{G}^{-\mathrm{T}}\dot{\boldsymbol{q}}, \boldsymbol{x}_{\mathrm{ref}}, \boldsymbol{0}, \boldsymbol{0}) = \tilde{\boldsymbol{r}} \quad \text{with} \quad \tilde{\boldsymbol{p}} = \boldsymbol{M}^{-1}\boldsymbol{p} \quad \text{and} \quad \tilde{\boldsymbol{r}} = \boldsymbol{M}^{-1}\boldsymbol{r}. \tag{30}$$

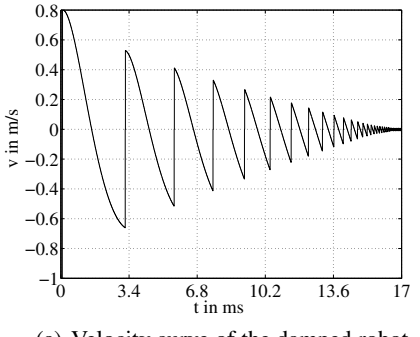
For verification purposes, the environment surface is assumed to satisfy

$$\Gamma(\boldsymbol{x}) = -z - z_0 = 0. \tag{31}$$

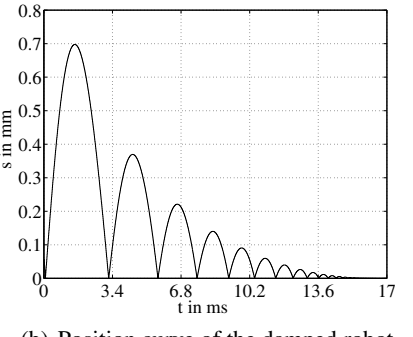
Hence, the unconstrained workspace of the damped robot is

$$L = \{\boldsymbol{x} \in \mathbb{R}^n : \Gamma(\boldsymbol{x}) < 0\}. \tag{32}$$

Fig. 6 shows the simulation results: A velocity controlled robot approaches the environment surface with $\dot{z} = -1\mathrm{m/s}$. As visualized in Fig. 6(a), the end effector is reflected and leaves the surface with $\dot{z} = 0.8\mathrm{m/s}$. Hitting the environment for a second time, the velocity is reduced to $\dot{z} = -0.66\mathrm{m/s}$. Hence, it can be shown that the subordinate drive control influences the bouncing behavior and finally establishes a stable contact, even if the environment shows purely elastic characteristics. The position is plotted in Fig. 6(b) and indicates that the end effector rests on the environment surface at $t = 17\mathrm{ms}$.



(a) Velocity curve of the damped robot



(b) Position curve of the damped robot

Fig. 6. Simulation of the transient phase with *Moreau's sweeping process*; parallel robot HEXA hitting a surface

Damping behavior of the system depends on the cycle time of the subordinate drive controller. The statements above hold for a simulation that is calculated with a cycle time of $T_s = 10\mu s$. The larger the cycle time of the subordinate drive controller the longer the damping phase until the end effector rests on the surface. Moreover, there is an upper threshold for the cycle time, where the subordinate drive controller cannot damp an elastic impact anymore. In this case, a stable transition phase can only be guaranteed if the environment has, at least partly, inelastic characteristics. In reality this situation can be assumed, because a real robot always possesses some elasticity that also contributes to this phenomena.

2.3.3 Stability Analysis

Stability of the implemented force control is influenced by the structure and parameters of the involved feedback loops. Taking this into account, the force control is stable if the following conditions are met:

(a) The end effector approaches the environment on the trajectory defined by the trajectory generator. Hence, the stability in this phase solely depends on the stability of the subordinate drive control (cf. [1]). If the drive controller is stable, this phase of the force controller is also stable.
(b) Stability during damping of the non-smooth impacts of the end effector on the environment surface is given by the stability of *Moreau's sweeping process*, cf. section 2.3.2 as well as eqs. (13) and (30). Here, the subordinated drive control dampens the post-impact velocity and thus establishes a stable contact.
(c) If a stable contact is established, the stability of the force controller designed in section 2.3.1 is crucial for the overall stability of the system. The synthesis is based on the phase-margin criteria and therefore inherently stable.
(d) If all above stability criteria are met, the force control is stable if the finite state machine, used to switch the controller modes according to the system states, always assures a defined final state representing the contact situation, cf. [12].

In addition to the stability criteria described before, the performance of the robot task depends on the Manipulation Primitive definition. This was first mentioned in [13] for the general case of a hybrid control scheme. Conflicting control objectives in various axes of the task frame may lead to a longer task execution time. Moreover, curved environments may exert forces not only in the normal axis, i. e. force controlled axis but also disturb the position control task in other directions. This has to be taken into account during robot programming.

Finally, it can be concluded that the force control scheme allows for a safe and stable execution of robot assembly tasks.

3 Evaluation and Experiments

Experimental setup for integrated force and motion control is given at hand for the robots HEXA and TRIGLIDE. The robot FIVEBAR is used for the evaluation of control schemes presented in [1]. General details can be found in [14, 15, 16] in

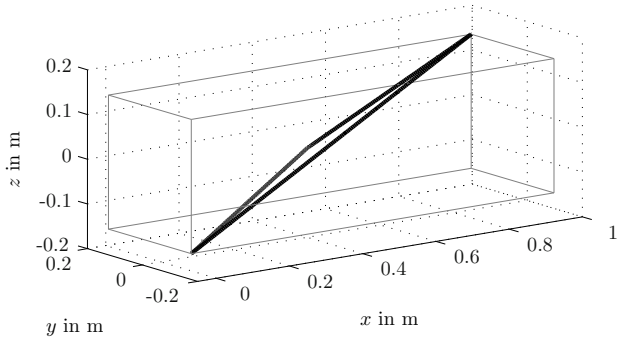


Fig. 7. Path driven on parallel robot TRIGLIDE

terms of comparison of control schemes, [17] with regard to influence of sliding mode on structural vibration and in [18] w.r.t. force control.

The performance of integrated control is shown w.r.t to aspects outlined in [1] – distortion of the trajectory in operational space, and section 2 – uniform force control and its performance in hybrid task specifications.

3.1 Trajectories on the TRIGLIDE

The performance of the control scheme concerning high dynamics was benchmarked on the TRIGLIDE-robot. The dynamic characteristics of the robot are presented in [19]. The trajectory was selected to cover a large area of the workspace and is shown in Fig. 7. The constraints are

$$(\text{jerk}, \text{acceleration}, \text{velocity}) = (4000\text{m/s}^3, 70\text{m/s}^2, 3\text{m/s}).$$

One criterion to evaluate the controller performance is the distortion of the trajectory in operational space, i. e. Cartesian space. Distortion here is defined by the deviation of the actual position from the reference trajectory

$$d(t) = \text{dist}\,(\boldsymbol{x}(t), g)\,, \tag{33}$$

where $\boldsymbol{x}(t)$ is the actual position in operational space and g is the set of all points of the path. Though $d(t)$ is a function of time t, it does not measure the time delay in trajectories. For the given path, a set g_i is defined for each section of the path as the straight line between the endpoints. The result is presented in the left column of Fig. 8. The distortion reaches its maximum value in line with maximum velocity and acceleration. The maximum distortion on the trajectory is $d_{\max} \approx 0.24$mm. The right column in Fig. 8 shows the effects of mass variation at the end effector. An additional mass of ($m = +2$kg) was attached to the end effector. According to analysis of parametric sensitivity (cf. Fig. 5 in [1]) this corresponds to $k_{T_v} > 1$ and thus decreases damping. Though the velocity error increases significantly, the path

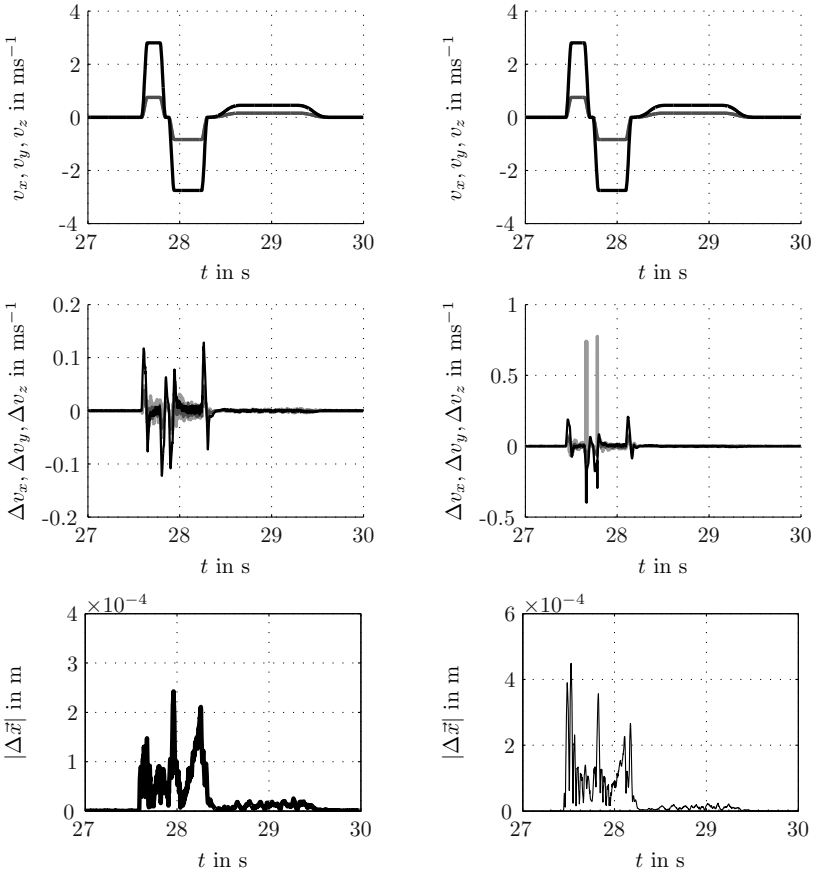


Fig. 8. Results on Triglide. The rows are associated with the velocity of planned trajectory, the velocity error and the distortion, respectively. The left column shows the nominal case, whereas the right column has a mass defect of Δm = +2kg at the end effector.

accuracy of the end effector is still high. The maximum difference to the trajectory is $d_{\max} \approx 0.45$mm.

3.2 Trajectories on the HEXA

The robot HEXA is used for evaluating the performance of force control as described in chapter 2 with special attention to hybrid control. The base frame is located in the middle of the plane spanned by the six drives.

In the first setup a block with dimensions (64 × 40) mm is misaligned by a Hybrid Move (HM)

$$^{\mathrm{TF}}\{x, y, z, \varphi_x, \varphi_y, \varphi_z\} = {}^{\mathrm{TF}}\{0.0\mathrm{m}, 0.0\mathrm{m}, 0.0\mathrm{m}, 5°, -10°, -15°\} \qquad \text{(HM:I)}$$

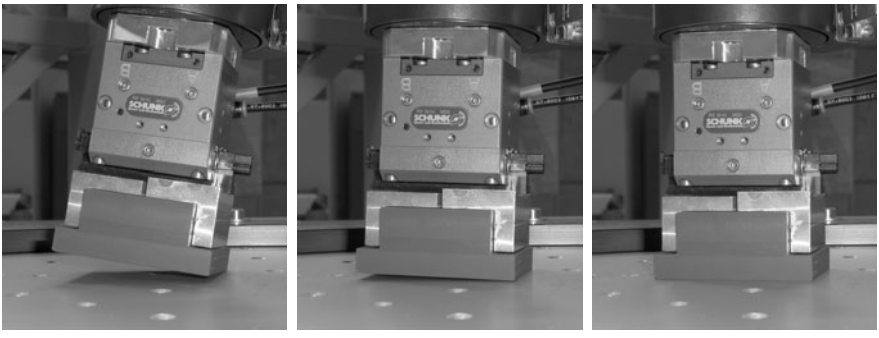


Fig. 9. Block placement by the HEXA-robot

in the task frame which is equal to the hand frame located at the tool center point with orientation parallel to the base frame. The task is to place the block on a flat surface with its normal vector parallel to the HEXA base frame. The distance to the surface is uncertain and is assumed to be approximately 0.03m. The evaluation process is depicted in Fig. 9. The force control consists of two control tasks: In a first step contact is established by hybrid force and position control with

$$^{\mathrm{TF}}\{x, y, f_z, \varphi_x, \varphi_y, \varphi_z\} = {}^{\mathrm{TF}}\{0.0\mathrm{m}, 0.0\mathrm{m}, -5.0\mathrm{N}, 0.0°, 0.0°, 0.0°\}. \qquad \text{(HM:II)}$$

Along with the task definition the constraints are set to

$$(\text{jerk}, \text{acceleration}, \text{velocity}, \text{impedance}) = (10.0\mathrm{m/s^3}, 1.0\mathrm{m/s^2}, 0.01\mathrm{m/s}, 500\mathrm{N/mm})$$

in the robots base frame. As stop condition a contact detection is specified.

In a second step the block is placed onto the surface by a Hybrid Move (HM)

$$^{\mathrm{TF}}\{x, y, f_z, \tau_x, \tau_y, \varphi_z\} = {}^{\mathrm{TF}}\{0.0\mathrm{m}, 0.0\mathrm{m}, -15.0\mathrm{N}, 0.0\mathrm{Nm}, 0.0\mathrm{Nm}, 0.0°\}, \qquad \text{(HM:III)}$$

where an execution time of $t = 12\mathrm{s}$ is given as stop condition. The above motion command bears coupled task specifications as the crank between the contact and task frame center point induces torques when applying force control. Hence, torque control to 0.0Nm can only be met if the conflict is solved by putting more weight to torque than to force control.

Experimental results are presented in Fig. 11. In Fig. 11(c) it can be seen that the robot moves toward the surface by specification of $f_z = -5.0\mathrm{N}$ in (HM:II). As the end effector is located in unconstrained space no significant force is measured. Therefore, the controller is internally switched to velocity control. Hence, the robot moves velocity controlled till $t \approx 7.0\mathrm{s}$, cf. Fig. 11(c). When touching the surface the damping phase is set up and applied as described in chapter 2.3.2. As the Hybrid Move stop condition is set to contact detection, the specification of $f_z = 5.0\mathrm{N}$ is not

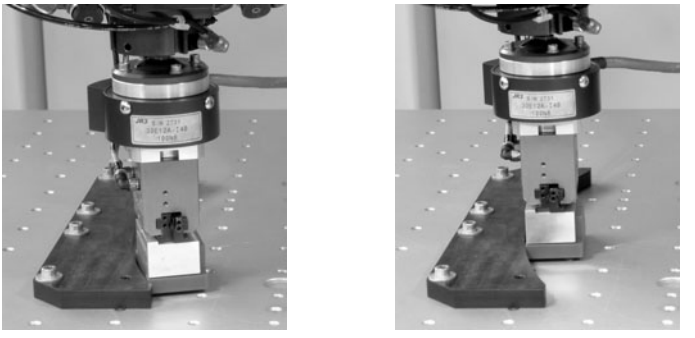


Fig. 10. Contour following task on HEXA-robot

used as command value for force control. A contact force of 7.5N is established. With stabilized contact the second Hybrid Move (HM:III) is executed at $t \approx 7.5$s, cf. Fig. 11(a) and (b). Here torque control is given more weight than force control due to coupling. As torque cannot directly be controlled because of unconstrained rotational degrees of freedom, the control task is switched to angular velocity control until the surface is hit, which occurs at $t_x \approx 15.75$s and $t_y \approx 16.6$s for x- and y-axis, respectively. At these points of time a switch back to torque control is executed. As there are no conflicting objectives anymore (no misalignment, effective crank equals zero), the reference force of $f_z = 15$N along the z-axis is established.

The second setup consists of contour following of a concave edge. The contact is established in a similar way to (HM:II) by two motions: a contact establishing with force control in the hand frame followed by an alignment to the contour using hybrid force and torque control

$$\begin{aligned} {}^{\mathrm{HF}}\{x, f_y, z, \varphi_x, \varphi_y, \varphi_z\} &= {}^{\mathrm{HF}}\{0.0\mathrm{m}, -5.0\mathrm{N}, 0.0\mathrm{m}, 0.0^\circ, 0.0^\circ, 0.0^\circ\} \\ {}^{\mathrm{TF}}\{x, f_y, z, \varphi_x, \varphi_y, \tau_z\} &= {}^{\mathrm{TF}}\{0.0\mathrm{m}, -15.0\mathrm{N}, 0.0\mathrm{m}, 0.0^\circ, 0.0^\circ, 0.0\mathrm{Nm}\}. \end{aligned} \quad \text{(HM:IV)}$$

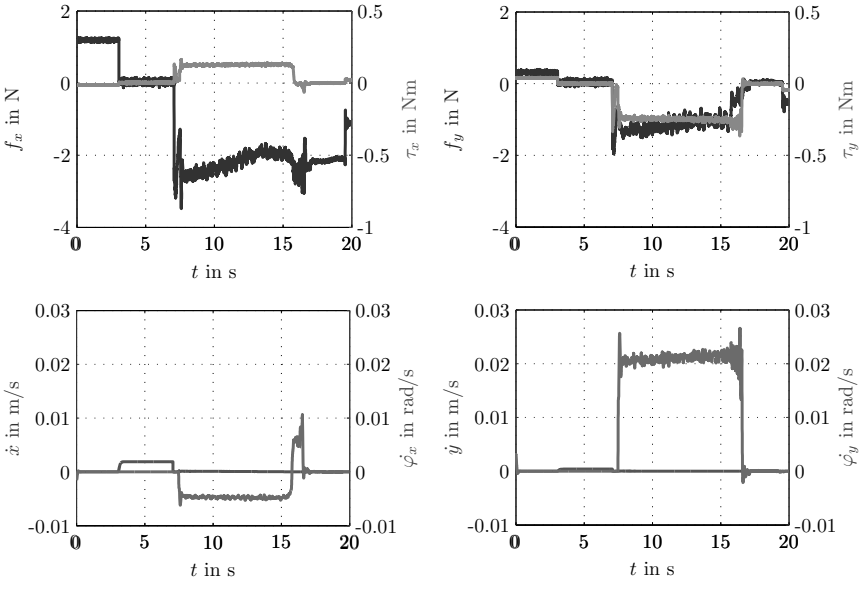
While the first movement is specified in hand frame coordinates, the second one is valid in the task frame which is set to the edge of the block ${}^{\mathrm{HF}}(-0.02\mathrm{m}, -0.03\mathrm{m}, 0.0\mathrm{m})$. As stop condition contact detection and establishment of specified force are used, respectively.

The contour following consist of a movement

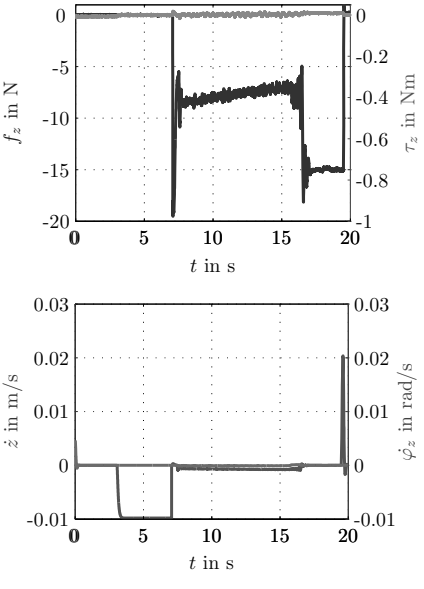
$$\begin{aligned} {}^{\mathrm{TF}}\{\dot{x}, f_y, z, \varphi_x, \varphi_y, \tau_z\} = \\ {}^{\mathrm{TF}}\{-0.01\mathrm{m/s}, -15.0\mathrm{N}, 0.0\mathrm{m}, 0.0^\circ, 0.0^\circ, 0.0\mathrm{Nm}\} \end{aligned} \quad \text{(HM:V)}$$

with the task frame located at ${}^{\mathrm{HF}}(0.0\mathrm{m}, -0.025\mathrm{m}, 0.0\mathrm{m})$. It stops when $\Delta x < -0.10$m and afterwards transfers the end effector away from the surface back to unconstrained space, cf. Fig. 10.

Experimental results in Fig. 12 present the robustness in hybrid control and thus reveal the advantage of the hierarchical motion force control scheme. During

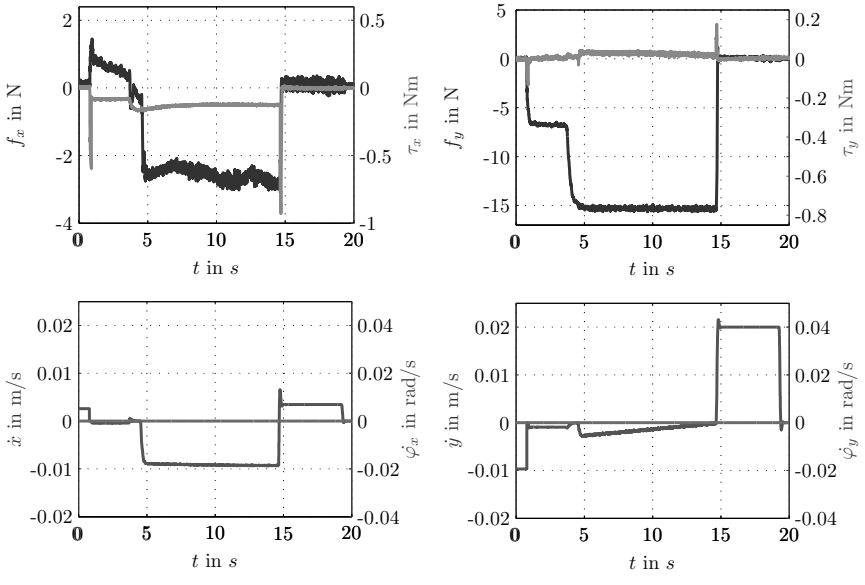


(a) x-direction (b) y-direction

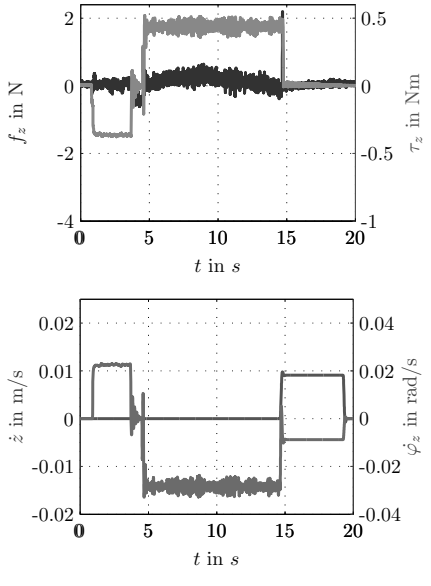


(c) z-direction

Fig. 11. HEXA block placement – upper row: forces (dark gray) and torques (light gray), lower one: translational (dark gray) and rotational (light gray) velocities.



(a) x-direction (b) y-direction



(c) z-direction

Fig. 12. HEXA contour following – upper row: forces (dark gray) and torques (light gray), lower one: translational (dark gray) and rotational (light gray) velocities.

hybrid motion (HM:V) the contact force of $f_y = -15$N (cf. Fig. 12(b)) guarantees a smooth movement along the concave profile without loosing contact as f_y does not drop to zero. However, the coupling arises a torque along the z-axis, cf. Fig. 12(c). Moreover, friction is induced in x-direction, see Fig. 12(a), and acts as disturbance for closed loop velocity control. Due to parametric design robustness is guaranteed, which can be verified by taking Coulomb friction law into account: the frictional force does not vary widely in x-direction, which means that a constant contact force and a constant tangential velocity must be given, which is proven by the plots given at hand. Hence both controllers offer robustness.

4 Conclusion

The proposed integrated control scheme matches the general requirements of the governing manipulation primitive concept. Special properties of parallel kinematic machines are considered and taken into account. Control on the drive level is used to encapsulate the robot dynamics and is developed for the purpose of high speed operation. Structural and parametric design has been paid special attention: Computed torque feed forward offers best performance for handling and assembly when parametric design is optimized for disturbance rejection along with the use of a cascaded structure.

Experimental results demonstrate the performance of integrated motion and force control in a hierarchical framework. Robustness is a keypoint in hybrid control, as friction, uncertainties of contact surfaces and of environmental system dynamics parameters and moreover conflicting objectives always have to be encountered in handling and assembly.

The design of the described force control concept bases on the trajectory interface of the underlying control scheme. It considers non-smooth mechanics during the transition phase to guarantee a stable installation of the contact. Consequently, robustness and performance are ensured for trajectory tracking in constrained and unconstrained space.

References

1. Kolbus, M., Wobbe, F., Reisinger, T., Schumacher, W.: Integrated Force and Motion Control of Parallel Robots – Unconstrained Space. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 233–252. Springer, Heidelberg (2010)
2. Hasegawa, T., Suehiro, T., Takase, K.: A model-based manipulation system with skill-based execution. IEEE Trans. Robot. Autom. 8(5), 535–544 (1992)
3. Raibert, M., Craig, J.J.: Hybrid position/force control of manipulators. ASME J. of Dyn. Systems, Measurement and Control 120(2), 126–133 (1981)
4. Hogan, N.: Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications. ASME Transactions Journal of Dynamic Systems and Measurement Control 107, 1–24 (1985)
5. Villani, L., Siciliano, B.: Robot Force Control (Kluwer International Series in Engineering & Computer Science), 1st edn. Springer, Berlin (2000) ISBN 9780792377337

6. Reisinger, T.: Kontaktregelung von Parallelrobotern auf der Basis von Aktionsprimitiven. Ph.D. thesis, TU Braunschweig (2008)
7. Kunze, M., Monteiro Marques, M.D.P.: An introduction to Moreau's sweeping process. In: Brogliato, B. (ed.) Impacts in Mechanical Systems: Analysis and Modelling, Euromech. Lecture notes in physics, pp. 1–60. Springer, Heidelberg (2000)
8. Acary, V., Brogliato, B.: Numerical Methods for Nonsmooth Dynamical Systems - Applications in Mechanics and Electronics. Lecture Notes in Applied and Computational Mechanics, vol. 35. Springer, Heidelberg (2008)
9. McClamroch, N.H., Wang, D.: Feedback stabilization and tracking of contrained robots. IEEE Trans. Autom. Control 33(5), 419–426 (1988)
10. Brogliato, B.: Nonsmooth Mechanics, 2nd edn. Springer, London (1999)
11. Maaß, J., Dietrich, F., Hesselbach, J.: RCA562: Control Architectures for Parallel Kinematic Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 315–331. Springer, Heidelberg (2010)
12. Assunco, J.M.V., Schumacher, W.: Hybrid force control for parallel manipulators. In: Proc. of the 11th Mediterranean Conference on Control and Automation (MED), Rhodos, Greece (2003)
13. Vukobratovic, M., Tuneski, A.: Contact control concepts in manipulation robotics: An overview. IEEE Trans. Ind. Electron. 41(1), 109–124 (1994)
14. Wobbe, F., Böske, W., Schumacher, W.: Optimierte Antriebsreglerstrukturen zur Störunterdrückung. In: SPS/IPC/DRIVES 2006, Nürnberg, Germany, pp. 473–482 (2006)
15. Wobbe, F., Kolbus, M., Schumacher, W.: Continuous sliding surfaces versus classical control concepts on parallel robots. In: Proc. of the 13th IEEE IFAC International Conference on Methods and Models in Automation, Szczecin, Poland (2007)
16. Wobbe, F., Kolbus, M., Schumacher, W.: Enhanced Motion Control Concepts on Parallel Robots. In: Automation and Robotics, pp. 17–40. In Tech Education and Publishing (2008) ISBN 9783902613417
17. Kolbus, M., Wobbe, F., Algermissen, S., Stachera, K., Schumacher, W., Sinapius, M.: Sliding mode control of a parallel robot with robust vibration control. In: Proc. of the 9th International Conference on Motion and Vibration Control (MoViC 2008), München (2008)
18. Reisinger, T., Kolbus, M., Wobbe, F.: Hybrid position/force-control of a planar parallel robot. In: Proc. of Conf. on Problems of Automated Electrodrives, Aluschta, Ukraine, pp. 250–254 (2005)
19. Schütz, D., Budde, C., Raatz, A., Hesselbach, J.: Parallel Kinematic Structures of the SFB 562. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 109–124. Springer, Heidelberg (2010)

Parallel Stiffness Actuators with Six Degrees of Freedom for Efficient Force/Torque Control Applications

Rafal Osypiuk and Torsten Kröger

Abstract. This chapter introduces stiffness actuators for six degrees of freedom (DOF) based on parallel kinematic structures. Robotic manipulators as well as common industrial environments usually feature high stiffnesses, and the transition from free space into stable contact is a demanding control task. The proposed stiffness actuators can be installed either at the manipulator's hand or in its environment. By applying active and highly dynamic impedance control, the maximum forces/torques between the manipulator and its environment caused by an impact can be significantly decreased, and cycle times in production can be greatly reduced. Beside theoretical derivations, this paper contains a number of real-world experimental results underlying the advantages of six-DOF stiffness actuators in the field of robot force/torque control.

1 Introduction

Force controlled interactions of robotic manipulators with their immediate environment are required for a huge number of tasks and applications of autonomous assembly, and have been the subject of intensive studies for nearly more than three decades — since the first concept of force control for industrial manipulators appeared [1]. Over the years, many interesting solutions have been developed that may

Rafal Osypiuk
Technische Universität Braunschweig, Institute for Robotics and Process Control,
Mühlenpfordtstraße 23, 38106 Braunschweig, Germany
and
Permanent address: West Pomeranian University of Technology in Szczecin,
Institute of Control Engineering, ul. 26 Kwietnia 10, 71126 Szczecin, Poland
e-mail: rafal.osypiuk@zut.edu.pl

Torsten Kröger
Stanford University, Artificial Intelligence Laboratory, Department of Computer Science,
Stanford, CA 94305-9010, USA
e-mail: tkr@stanford.edu

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 275–291.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

be divided into two general groups: indirect and direct force control, exemplified by hybrid force/pose control [2], impedance control [3], model-based control [4, 5], adaptive control [6, 7], passivity control [8], etc. The use of six-degree-of-freedom (DOF) force/torque sensors opens the doors to defining and executing compliant robot motions by means of the above-mentioned systems. However, it should be kept in mind that the manipulator itself represents a kinematic chain of high stiffness, and the above algorithms only imitate the natural compliance. Therefore, the mentioned systems are applicable only within a narrow range of low frequencies, and they do not solve the problem of transitions from free space to contact in case of high-stiffness environments. This is a serious problem, because the measured force/torque signals show undesired, jerky behavior during the impact to the environment [9]. This common behavior leads to undesired strains of the manipulator and the manipulated object(s) or of the environment itself. In order to prevent high impact forces and torques, the manipulator's velocity during expected contact transitions often becomes limited — which entails limited effectiveness during force/torque controlled processes.

The above problems can be resolved by using compliant systems, which may be accomplished in Cartesian space [10, 11] or in joint space [12]. The first option may be exemplified by the well-known concept of Remote Center of Compliance (RCC) [13]. This is a very simple mechanical solution based on springs or elastic materials. RCC compensates small position errors during compliant manipulator motions. Disadvantages of such a solution are the lack of information about the exact position of the gripper in space, a very small workspace, and usually a constant compliance defined by the properties of the mechanical components. More promising but also much more complicated for solving this problem is the use of a passive compliance elements in the robot joints. This leads to new solutions called compliant robots. Such systems are commercially available [14, 15, 16]. To obtain a natural compliance, mechanical elastic couplings or the effect of air compressibility (hybrid pneumatic-electric motors) are employed within these systems. This way, it becomes possible to compensate force/torque peaks during contact transitions. The disadvantage of this solution is its great mechatronic complexity, which calls for advanced control systems [17].

Such compliant robots present an interesting way for manipulators that should be safe in their specific environment, for example, service robots cooperating with human beings. However, these robots are hardly suited for many industrial compliant motion tasks such as robot assembly. When these machines collide with high-stiffness environments, substantial force/torque impulses appear. A more promising solution seems to be the realization of an active compliance in Cartesian space, which significantly reduces the effective manipulator stiffness, hence, resulting in a substantial decrease of forces/torques during contact transitions. In this paper, a new concept of stiffness actuators based on a parallel kinematic structure is presented: A HEXA platform is mounted on the environment and a Gough/Stewart platform is mounted on the end-effector of the manipulator. The obtained test results provide unambiguous evidence in favor of the proposed solutions; they show that the employment of the considered systems provides substantial benefits. These benefits are

a significant decrease of maximum impact forces/torques, the possibility of a great increase of approach velocities before expected contact transitions, and, hence, a significant decrease of execution times of compliant motions tasks (e.g., assembly tasks).

2 Addressed Force/Torque Control Problems

Talking about control, we are usually interested in the properties of the *overall process*. A six-joint industrial manipulator is a complex multi-input/multi-output system characterized by strong nonlinear and time-varying behaviors. In addition, if a manipulator comes into environmental contact, the problem of time-variant environmental stiffness behavior may arise. In such cases, the stiffness may fluctuate even 500-fold [18] leading to changes of the gain of the force control loop. This fact can have a dramatic effect on the settling time, if a classic low-robust PID system is applied [18].

Another phenomenon is the problem of transitions from free space to contact in case of high-stiffness environments. Industrial manipulators feature high end-effector stiffnesses resulting in excellent positioning accuracies. However, problems arise when the robot is used for force control applications. In such cases, the high stiffness may account for the generation of undesired and high contact forces/torques. Figure 1 shows impact forces in dependency of the Cartesian velocity of the manipulator in approach direction. For example, at a linear velocity in z-direction of 25 mm/s, the respective contact force achieves a value of more than 250 N.[1]

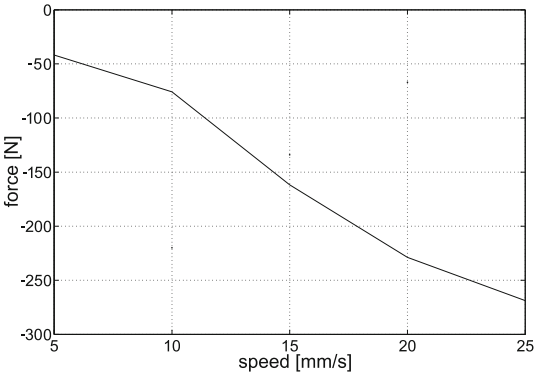


Fig. 1. High-stiffness environment contact with different Cartesian speeds.

For this reason, passive and/or active compliance systems are used to improve the operational quality of force/torque control. A passive compliance module is commonly composed of some mechanical devices of springs or knee joints — there is no interaction between the control architecture of the manipulator and the compliant device. As a result of the above-mentioned disadvantages, passive compliance

[1] Performed with a Stäubli RX60 industrial manipulator [19].

devices do not lead to significant improvements, whereas active systems are of great relevance here. Active compliance systems employ sensors, usually six-DOF force/torque sensors to interact with the robot motion controller [20]. However, these kinds of solutions call for advanced control systems in order to cope with critical stability conditions [21].

3 Natural Compliance versus Stiffness Actuators

Contacting the environment with a robotic end-effector presents a very complex phenomenon that is difficult to evaluate from a mathematical point of view [9, 22]. For the description of this process, one can consider simple static models that take account of the environment stiffness [23] or more advanced mathematical descriptions that allow for intricate dynamic phenomena [24]. In [20, 25], a modular model is proposed, which can be extended in terms of reflecting only static properties to reflecting dynamics, too. To determine the difference between the natural and artificial stiffness, a simple linear Kelvin-Voigt model and its nonlinear generalization the Hunt-Crossley model [26] was used. A simple analysis in time domain (Fig. 2) and frequency domain (Fig. 3) has been carried out for the determined models. The analysis unambiguously demonstrates the differences between the compliances under its consideration.

The phase of contacting the environment with natural and artificial compliance by the manipulator has been simulated first. It is essential that both environments feature an identical low stiffness. As illustrated in Fig. 2, establishing the contact with the natural compliance could be processed smoothly. On the other hand, contacting the artificial compliance (actuator) having the same stiffness lead to violent force variations, which is due to the need of accelerating the moving parts of the actuator (i.e., the mass of which is obviously non-zero).

The differences between the natural and artificial compliance are much more highlighted by the frequency response shown in Fig. 3. Static properties of the natural compliance make its value independent of the frequency, which can not be said

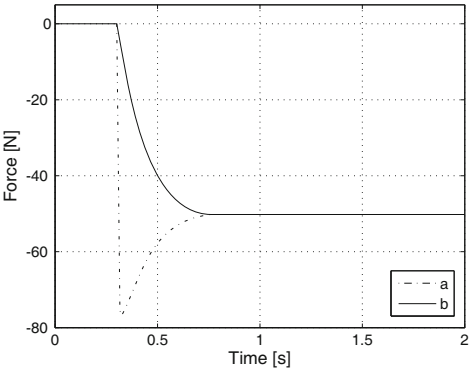


Fig. 2. Simulation of establishing the contact with different kinds of environment, but of the same stiffness: **(a)** stiffness actuator (artificial compliance); **(b)** natural compliance.

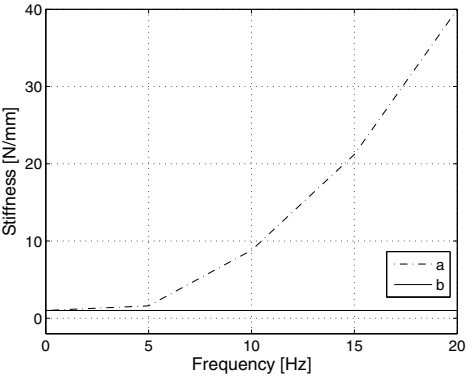


Fig. 3. Change in stiffness at periodically varied displacement of the manipulator permanently contacting the environment: **(a)** stiffness actuator (artificial compliance); **(b)** natural compliance.

about the artificial compliance as a dynamic system. The actuator correctly emulates the natural compliance for low frequencies. To ensure its desirable operation over a wider range of frequencies, the mass of the moving parts of the actuator should be as small as possible.

4 Stiffness Actuator Installed in the Environment of the Manipulator

In the following, the contact between two stiff bodies, i.e., between a manipulator and its environment, is regarded. In this case, it is irrelevant whether the compliant part is installed in its environment or mounted on the end-effector of the manipulator. But: these two cases feature different properties, and they require different assumptions for their realization. Hence, we discuss the first solution in this section, and the second solution in the next section.

To construct the stiffness actuator of Fig. 4, a closed kinematic structure was chosen intuitively. Rotational drives are employed, which leads to the well-known

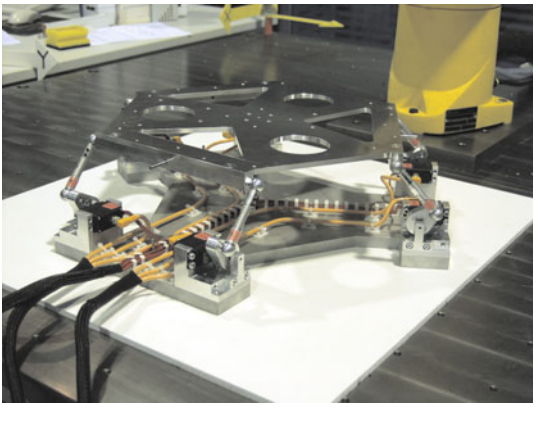


Fig. 4. The realized stiffness actuator (HEXA platform).

HEXA construction design in the six-DOF case [27]. Undoubtedly, a proper kinematic structure may simplify the mathematical description (kinematics, dynamics, etc.) [28] and ease the control over the entire actuator operation space. In case of closed kinematic chains, it should be remembered that control in Cartesian space solely seems to be worthwhile. Hence, it is essential that the arrangements of drives, selection of joint lengths, and dimensions of the movable platform in relation to the robot base are accomplished in order to prevent singularities over the entire workspace of the actuator [29, 30]. It is essential that the movable masses of the HEXA platform are as small as possible without sacrificing the stiffness properties of the system. All mechanical components of the manipulator are made of aluminum (including the base and the end-effector platform of the actuator). In addition, finite element methods have been applied in the COSMOS™package of SolidWorks™ [31] to design the HEXA platform with the aim to reduce its mass and to admit loads as high as possible. A quite limited workspace and a lack of stringent requirements on accuracy and repeatability of positioning has made it possible to employ simple servo drives[2] to drive the HEXA platform. This is a robot-dedicated solution, which is based on high-dynamic, coreless motors and titanium gears. It should be stressed here that the control units delivered by the manufacturer of the servo systems has been removed and replaced by an in-house solution in order to achieve the required control performance.

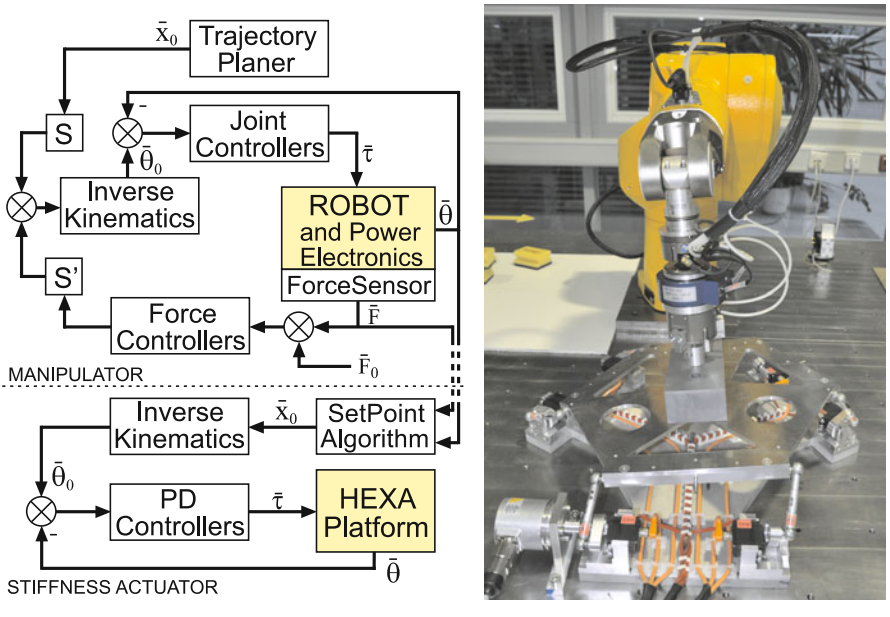


Fig. 5. Experimental setup with stiffness actuator installed in the environment.

[2] Type of the servo drives: HSR-5995TG, [32].

The stiffness actuator (Fig. 4) is controlled in such a way that the robot manipulator "feels" a constant environmental stiffness irrespective of any object properties, i.e., there is no difference if a steel or a rubber element is contacted. This mechatronic solution gives almost one hundred percent robustness against environmental stiffness variations, such that a significant shortening of execution times during transitions from free space to contact can be easily achieved. Figure 5 depicts the overall experimental setup to realize force/torque control systems based on the approach of such an active environment. The setup includes the Stäubli™RX60 industrial manipulator [19], a JR3™force/torque [33] sensor, and the six-DOF HEXA platform as an stiffness actuator. Both, the manipulator and the stiffness actuator, make use of the open robot control architecture based on MiRPA (Middleware for Robotic and Process Control Applications, [34, 35]), which enables a high-performance and transparent real-time information flow between all required software components. This is a typical mechatronic solution, where no advanced control systems are employed, but only classic PID control. Regarding the control of the manipulator, a hybrid force/pose control scheme is implemented [2, 36, 25, 37, 38], whereas the axes of the stiffness actuator are controlled independently.

The hybrid force/pose control system of the robotic manipulator does not obtain any feedback information about the active environment. The HEXA platform emulates the environment's elasticity in two ways:

- passively, i.e., without any information about the acting force;
- actively driven by the manipulator's force/torque and pose feedback.

The former solution is accomplished by means of a fixed set-point control and appropriately chosen gain values in the position control loops. Hence, the platform stiffness can be defined over a wide range.

The latter approach makes use of information on the values of the contact forces/torques and the manipulator pose in order to generate the motion trajectory of the HEXA platform on-line. This approach is much more sophisticated in terms of control, mainly due to stability conditions.

Table 1. Advantages and disadvantages of using a stiffness actuator installed in the environment of the manipulator.

Advantages	Disadvantages
• The compliance of the manipulator remains uninterfered. • The tool positioning accuracy remains unchanged. • A greater latitude in the choice of platform drives and geometry can be achieved.	• The mass of the movable part of the actuator is significant. • The platform stiffness is dependent on the location of the point of contact. • Only objects of quite small masses can be manipulated. • The manipulator workspace becomes diminished due to the installation of the stiffness actuator. • The universality may be inconsiderable.

Although, as will be shown in Sec. 6, employing an active environmental compliance offers many advantages, the system is not free from some significant limitations (cf. Table 1). The most important of them is that the platform stiffness is dependent on the location of the point of contact, which requires to accomplish a mating operation in the close vicinity of the actuator center. Furthermore, the masses of objects to be manipulated are subject to substantial restrictions, which additionally detracts the universality of such a solution. The mentioned limitations can be eliminated by employing a further stiffness actuator mounted on the end-effector of the manipulator (cf. Sec. 5).

5 Stiffness Actuator Mounted on the End-Effector of the Manipulator

As noted above, the mass of the movable part of the stiffness actuator should be as small as possible in order to achieve a compliance as close as possible to the natural one. This ensures small impact forces/torque values at the instant the manipulator is brought into contact with its environment. This condition can be sufficiently fulfilled if an active compliance system is installed at the manipulator's wrist. Here, we made use of the well-known Gough parallel kinematic structure [39], often mistakenly called *Stewart Platform* [28]. The actuator is based on six classical pneumatic drives (Fig. 6) allowing to obtain a desired compliance by the compressibility of air. Obviously, the arrangement of drives and the platform geometry should lead to a complete lack of singularities over the entire workspace of the actuator, which — in our case — allows translational motions of: $\hat{X} = \pm 28\,\text{mm}, \hat{Y} = \pm 28\,\text{mm}\,\hat{Z} = \pm 12.5\,\text{mm}$ as well as rotational motions of: $\alpha_x = \pm 22^\circ\, \alpha_y = \pm 22^\circ\, \alpha_z = \pm 34^\circ$.

Fig. 7 depicts the control scheme of the stiffness actuator mounted on the wrist of the manipulator (including the force/torque control loops). Compared to Fig. 5,

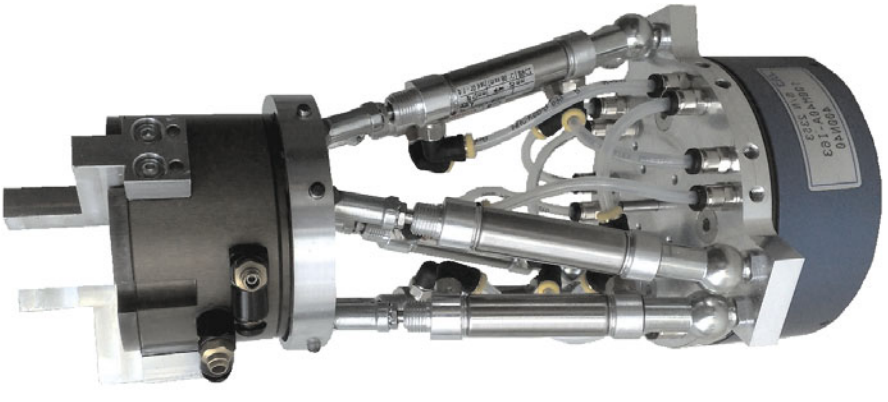


Fig. 6. Pneumatic Gough platform together with the gripper and the JR3 force/torque sensor mounted on the end-effector of the manipulator.

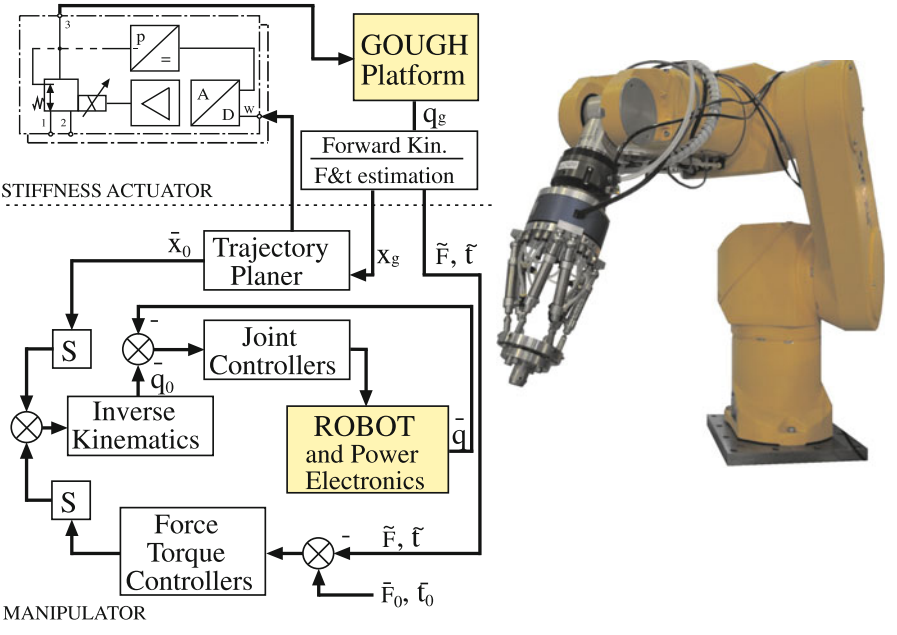


Fig. 7. Experimental setup with the stiffness actuator mounted on the wrist of the manipulator.

the changes are insignificant. The actual gripper pose from the information delivered by the forward kinematics block of the Gough platform is employed here. In addition, the trajectory generation module is responsible for servicing the symmetrically driven pneumatic cylinders. Two principal actuator operation modes can be distinguished here:

(a) If the manipulator moves in free space, and the chance of colliding with some environmental object is negligible, the pistons are in their most rear position in order to prevent undesired vibrations of the gripper and/or the object in front of the platform.
(b) If a collision of the end-effector and the environment is expected, the control system opens proportional pressure valves to position the pistons at about 50 percent of their operating range. This way, the actuator platform is prepared for a transition from free space to environmental contact.

Except for the actuator initialization phase, in which the pistons are set to the position for expected collisions, an open-loop controller is employed in the system. The position-dependent feedback is used primarily as information about the actual gripper pose in the space, which is applied to the manipulator controller after the forward kinematic model is recalculated. This is one of the weaknesses exhibited by the active compliance module if installed on the wrist of the manipulator: the detrimental effect on the positioning accuracy of the overall system. However, it should be remembered that the positioning accuracy in closed kinematic chains is at least two orders higher [28] than that in serial structures, where the problem of

Table 2. Advantages and disadvantages using a stiffness actuator mounted on the wrist of a robot manipulator.

Advantages	Disadvantages
• Only small masses are in motion, which significantly reduces force/torque peaks during impacts. • Compliance is always realized w.r.t. the platform center. • Objects of large masses can be manipulated. • Straight-forward integration in the manipulator control system is possible. • Compact design. • Very high system flexibility. • Possibility to estimate contact forces/torques from stiffness matrices.	• Adverse effect on positioning accuracy (insignificant). • Necessity to deliver compressed air and to provide wiring to the sensors of the active platform. • Relatively sophisticated mechanical structure.

error additivity exists. Therefore, the problem may be regarded as rarely significant. In Table 2, the advantages and disadvantages of the actuator mounted on the end-effector of the manipulator are shown. As a result, the actuator concept presented in this section clearly leads to important benefits as far as an active compliance system is also installed in the manipulator's environment (cf. Table 1).

6 Experimental Results

In the next subsections, real-world experimental results achieved with the considered stiffness actuators are presented. In all cases, the open robot control architecture based on MiRPA was used for the experiments. It was possible to achieve a control rate of 1 kHz. A rapid control prototyping system based on Matlab/Simulink™ [40] and Opal RT-Lab™ [41] was used, such that all controllers essential for this set-up could be developed in a very flexible way.

6.1 Robustness

If the manipulator comes into contact with a high-stiffness environment, the gain of the force control loop will become very high. To provide stability, the force controller gain should be considerably decreased. This, however, adversely affects the robot dynamics. If the manipulator has to deal with environments of varying stiffnesses (e.g., in the process of advanced assembling), then it will be essential to employ robust control systems [18] in order to provide a stable and time-optimal control solution. Otherwise, the time needed to accomplish a task will be unacceptably high.

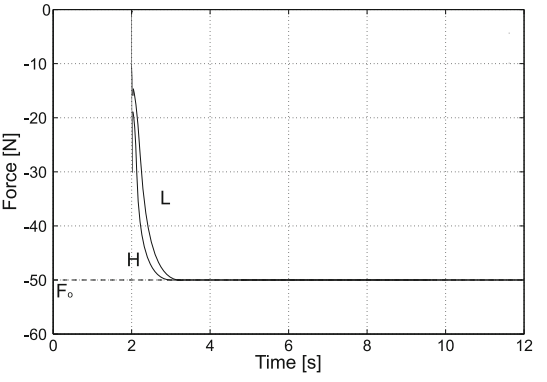


Fig. 8. Results of the force control experiment with two different environmental stiffnesses: The classic case (i.e., without an active compliance module).

In such a case, it may be expedient to employ an active compliance system between the manipulator and the environment, which makes it possible to change the effective manipulator stiffness. By this means, a significant increase of robustness against stiffness variations while establishing the contact may be achieved. Furthermore, this provides an unchanged settling time.

Results of a simple test are given below, and Figs. 8 and 9 depict results of a force control experiment for the one-DOF case. Here, the major attraction is given to the contact of the robot with environments of two different stiffnesses, i.e., an aluminum plate of high stiffness (H) and a PVC plate of low stiffness (L). Both cases have been tested for the classic situation (i.e., the manipulator without any compliance equipped with a force/torque sensor driven by a hybrid force/pose controller) and the manipulator with an actively controlled compliance (equipped with a force/torque sensor, hybrid force/pose control, and stiffness actuator).

The way, in which the controllers have been parameterized, deserve further comments. The controllers were tuned in the both cases only once (control with fixed set-point) for one chosen point of operation. In the classic case (Fig. 8), this operation point was determined by the contact with environment of high stiffness[3]. In the second case (Fig. 9), the controllers were tuned for the chosen actuator compliance, and they were not adapted in any further way.

In Fig. 8, a deterioration in the control performance may be noticed. This is due to the changed environment stiffness, which also manifests itself in the changed gain of the force control loop. By using an active compliance (cf. Fig. 9), this phenomenon is not visible, because the manipulator "feels" an unchanged environmental stiffness, regardless of the properties of objects to be manipulated. Hence, the settling time remains unchanged. In addition, Fig. 9 illustrates the above-mentioned phenomenon, i.e., an impulse-like change of the force value at the instant the contact with the stiffness actuator (here: the HEXA platform) is established. The maximum force value during the impact is determined (amongst others) by the mass of the moving part of the stiffness actuator.

[3] To ensure a stable force control when the manipulator comes into contact with environment of different stiffness, it is required that the PID controller is tuned for contacts with environment of high stiffness (high control plant gain, small controller gain) [18].

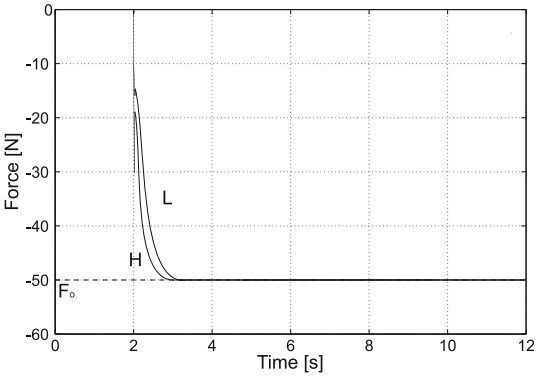


Fig. 9. Results of the force control experiment with two different environmental stiffnesses: Using a stiffness actuator.

6.2 Peg-in-Hole Task

To show the benefits offered by stiffness actuators, a general test, in which all six DOFs are involved, has been carried out: the well-known peg-in-hole task. This sophisticated task calls for an advanced decision-making system based on force/torque control. Without going into details, two different situations are regarded: accomplishing the peg-in-hole task with and without an active environment. As a result, the overall execution time and the accompanying averages of force/torque values are significantly different.

In the experiment, the HEXA platform served as the stiffness actuator in its simplest configuration, i.e., as a passive system and without any force/pose dependent feedbacks. To this end, we made use of the properties exhibited by PD position controllers of the platform (i.e., the PD controllers exhibit a steady-state error when an external force is acting). Figure 10 depicts a plot of the platform stiffness in z-direction for different gains of the platform position controllers. The visible nonlinearity is a consequence of that the motion is performed by means of rotary drives. As can be seen, this is a simple way to determine the actuator stiffness; it can be modified even during the control process (if required).

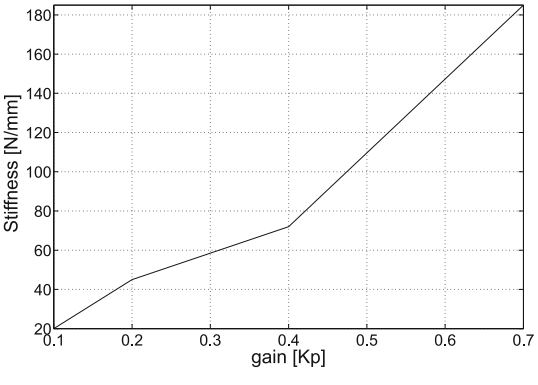
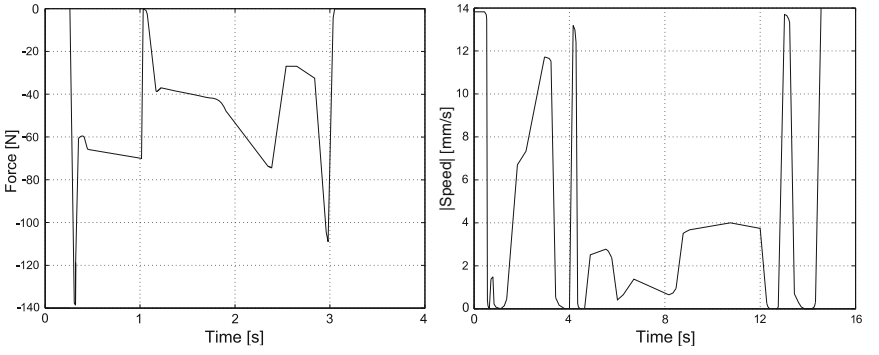
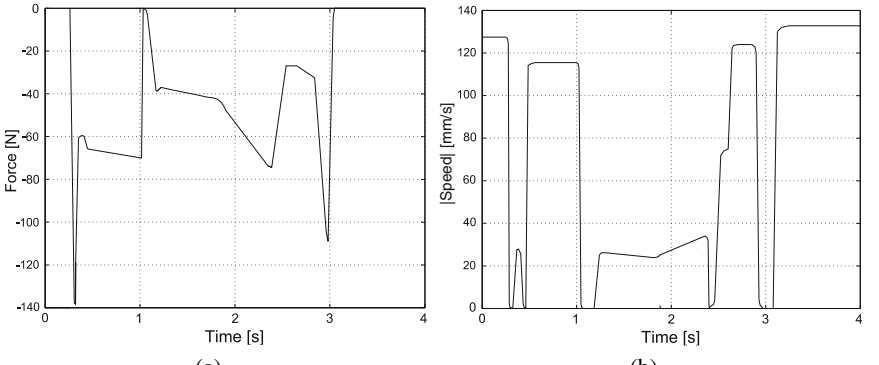


Fig. 10. Platform stiffness (in z-direction) in dependence of the controller gain.



(a) (b)

Fig. 11. (a) Average forces and (b) approach velocities vs. time during the peg-in-hole task without active compliance.



(a) (b)

Fig. 12. (a) Average forces and (b) approach velocities vs. time during the peg-in-hole task with the stiffness actuator installed in the environment.

Figures 11 and 12 show the execution times of the peg-in-hole task in two cases: without active compliance and with the stiffness actuator. The maximum time is influenced by the limits imposed on the force/torque values ($F_{x,y,z} = 150\,N$, $\tau_{x,y,z} = 10\,Nm$). Due to the increased dynamics of force/torque controllers and due to increased manipulator approach velocities (figs. 11 and 12) when contacting the environment, the overall execution times of such tasks can be significantly reduced. A video presentation on how the realized stiffness actuator works can be found in [42].

6.3 Contact Forces

The phase of establishing the contact between the manipulator and its environment presents a fundamental challenge to robot motion control. The manipulator

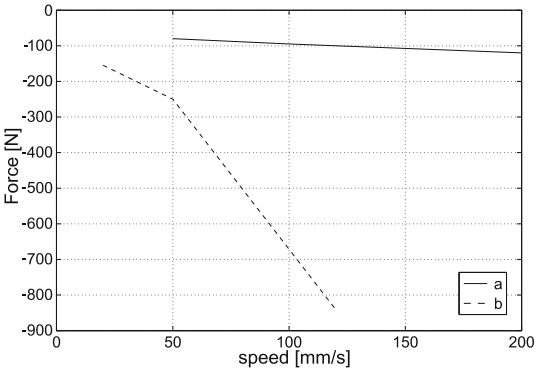


Fig. 13. Contact to high stiffness environments with different approach velocities. **(a)** Actuator mounted on the manipulator's end-effector. **(b)** Without active compliance.

is a kinematic chain of great mass, and it is not able to stop instantaneously at the moment of contact detection. This may account for great forces/torques generated when contacting an environment of high stiffness, even at quite small Cartesian velocities. This circumstance requires a drastic reduction of manipulator approach velocities just before the expected instant of collision. In case of unstructured environments, reasonable force/torque control is not possible from a practitioner's point of view. The solution of such problems significantly benefits from the employment of actively controlled compliance systems as presented in this paper.

As a further option, the stiffness actuator can be mounted on the end-effector, which ensures the properties summarized by Table 2. To clarify the benefits offered by such a solution, a simple experiment was carried out: contacting an environment of high stiffness at different approach velocities (motion performed only in Cartesian z-direction; normal to the surface). A Stäubli RX90 industrial manipulator was equipped with and without the pneumatic actuator shown in Fig. 7. As can be seen in Fig. 13, the proposed solution provides a tenfold reduction of contact forces. This reduction substantially improves the effectiveness of force/torque control approaches in general.

7 Conclusions

The above-presented method of stiffness actuators offers a very interesting extension to many existing methods of force/torque control known from literature. The use of stiffness actuators enables us to control the effective stiffness of robotic manipulators in Cartesian space; thus, solving the problem of establishing stable contacts with stiff environments. In addition, approach velocities can be greatly increased, maximum and average force/torque values during transitions from free space to contact are reduced, and — finally — execution times of force/torque controlled manipulator-environment interactions become decreased.

Acknowledgements. The authors are deeply indebted to the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), whose funding of the Sonderforschungsbereich 562 (SFB, Collaborative Research Center 562) made this work possible. Furthermore, the authors would like to thank *QNX Software Systems* for providing free software licenses.

References

1. Whitney, D.E.: Force feedback control of manipulator fine motion. ASME Journal of Dynamic Systems, Measurment and Control 98, 91–97 (1977)
2. Raibert, M.H., Craig, J.J.: Hybrid position/force control of manipulators. ASME Journal of Dynamic Systems, Measurement and Control 102, 126–133 (1981)
3. Hogan, N.: Impedance control: An approach to manipulation. Part I: Theory. Part II: Implementation. Part III: Applications. ASME Journal of Dynamic Systems, Measurment, and Control 107, 1–24 (1985)
4. Massoud, A.T., EIMaraghy, H.A.: Model-based motion and force control for flexible-joint robot manipulators. The International Journal of Robotics Research 16(4), 529–544 (1997)
5. Osypiuk, R., Finkemeyer, B.: Hybrid model based force-position control: Theory and experimental verification. Robotica 24(6), 775–783 (2006)
6. Roy, J., Whitcomb, L.L.: Adaptive force control of position/velocity controlled robots: Theory and experiment. IEEE Trans. on Robotics and Automation 18(2), 121–137 (2002)
7. Kröger, T., Finkemeyer, B., Heuck, M., Wahl, F.M.: Adaptive implicit hybrid force/pose control of industrial manipulators: Compliant motion experiments. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, pp. 816–821 (2004)
8. Siciliano, B., Villani, L.: A passivity-based approach to force regulation and motion control of robot manipulators. Automatica 32(3), 443–447 (1996)
9. Brogliato, B.: Nonsmooth Mechanics. In: Communications and Control Engineering. Springer, London (1999)
10. Joo, S., Waki, H., Miyazaki, F.: On the mechanics of elastomer shear pads for remote center compliance (rcc). In: Proc. of the IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, vol. 1, pp. 291–298 (1996)
11. Denkena, B., Wedler, A., Friederichs, O.J., Hackbarth, A., Hackelöer, F.: New compliant hexapod tool for industrial robotic applications driven by pneumatic mckibben actuators. In: Eleventh International Conference on New Actuators and Fifth International Exhibition on Smart Actuators and Drive Systems, Bremen, Germany, pp. 153–156 (2008)
12. Albu-Schäffer, A., Ott, C., Frese, U., Hirzinger, G.: Cartesian impedance control of redundant robots: Recent results with the DLR-light-weight-arms. In: Proc. of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, vol. 3, pp. 3704–3709 (2003)
13. Whitney, D.E., Nevins, J.L.: What is the remote center compliance (RCC) and what can it do. In: Proc. of the ninth International Symposium on Industrial Robots, Washington, D.C., USA, pp. 135–152 (1979)
14. KUKA Roboter GmbH, Zugspitzstraße 140, D-86165 Augsburg, Germany (2008), `http://www.kuka.com/en/company/group` (accessed: December 15, 2008)
15. Mitsubishi Heavy Industries, Ltd., Mitsubishijuko Yokohama Bldg., 3-1, Minatomirai 3-chome, Nishi-ku, Yokohama, 220-8401, Japan (2008), `http://www.mhi.co.jp/en` (accessed: December 15, 2008)
16. FerRobotics Compliant Robot Technology GmbH, Hochofenstraße 2 (Voest BG 01), A-4030 Linz, Austira (2009), `http://www.ferrobotics.at` (accessed: July 31, 2009).
17. Luca, A., Albu-Schäffer, A., Haddadin, S., Hirzinger, G.: Collision detection and safe reaction with the DLR-III lightweight manipulator arm. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, pp. 1623–1630 (2006)

18. Osypiuk, R., Kröger, T.: A three-loop model-following control structure: Theory and implementation. International Journal of Control 83, 97–104 (2010)
19. Stäubli Faverges SCA, Place Robert Stäubli BP 70, 74210 Faverges (Annecy), France (2008), http://www.staubli.com/en/robotics (accessed: December 15, 2008)
20. Reisinger, T.: Kontaktregelung von Parallelrobotern auf der Basis von Aktionsprimitiven Interaction Control of Parallel Robots Based on Skill Primitives. Ph.D. thesis, Institut für Regelungstechnik, Technische Universität Carolo-Wilhelmina zu Braunschweig (2008), http://www.digibib.tu-bs.de/?docid=00022368 (accessed: December 15, 2008)
21. Wen, J., Murphy, S.: Stability analysis of position and force control for robot arms. IEEE Trans. on Automatic Control 36(3), 365–371 (1991)
22. Featherstone, R., Thiebaut, S.S., Khatib, O.: A general contact model for dynamically-decoupled force/motion control. In: Proc. of the IEEE International Conference on Robotics and Automation, Detroit, MI, USA, vol. 4, pp. 3281–3286 (1999)
23. Sciavicco, L., Siciliano, B.: Modelling and Control of Robot Manipulators. In: Advanced Textbooks in Control and Signal Processing, 2nd edn. Springer, Heidelberg (2000)
24. Mills, J.K., Lokhorst, D.M.: Stability and control of robotic manipulators during contact/noncontact task. IEEE Trans. on Robotics and Automation 9(3), 335–345 (1993)
25. Reisinger, T., Kolbus, M., Wobbe, F., Schumacher, W.: Integrated Motion and Force Control of Parallel Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 233–252. Springer, Heidelberg (2010)
26. Diolaiti, N., Melchiorri, C., Stramigioli, S.: Contact impedance estimation for robotic systems. IEEE Trans. on Robotics 21(5), 925–935 (2005)
27. Pierrot, F., Dauchez, P., Fournier, A.: Fast parallel robots. Journal of Robotic Systems 8(6), 829–840 (1991)
28. Merlet, J.-P.: Parallel Robots. In: Solid Mechanics and Its Applications, 2nd edn., Springer, Heidelberg (2006)
29. Hesselbach, J., Bier, C., Campos, A., Löwe, H.: Direct kinematic singularity detection of a hexa parallel robot. In: Proc. of the IEEE International Conference on Robotics and Automation, Washington, D.C., USA, pp. 3238–3243 (2005)
30. Bier, C., Pietsch, I., Maaß, J., Hesselbach, J.: Detection and Avoidance of Singularities. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 77–92. Springer, Heidelberg (2010)
31. Dassault Systèmes SolidWorks Corp., 300 Baker Avenue Concord, MA 01742, USA (2009), http://www.solidworks.com (accessed: August 6, 2009).
32. Hitec RCD Korea, Inc., 653, Yangcheong-Ri, Ochang-Myeon, Cheongwon-Gun, Chungcheongbuk-Do, Korea (2009), http://hitecrcd.co.kr (accessed: August 6, 2009)
33. JR3, Inc., 22 Harter Ave, Woodland, CA 95776, USA (2008), http://www.jr3.com (accessed: December 15, 2008)
34. Finkemeyer, B., Kröger, T., Kubus, D., Olschewski, M., Wahl, F.M.: MiRPA: Middleware for robotic and process control applications. In: Workshop on Measures and Procedures for the Evaluation of Robot Architectures and Middleware at the IEEE/RSJ International Conference on Intellegent Robots and Systems, San Diego, CA, USA, pp. 78–93 (2007)
35. Finkemeyer, B., Kröger, T., Wahl, F.M.: A Middleware for High-Speed Distributed Real-Time Communication. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 193–212. Springer, Heidelberg (2010)

36. Villani, L., Schutter, J.D.: Force control. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics, 1st edn., ch. 7, pp. 161–185. Springer, Heidelberg (2008)
37. Maaß, J., Dietrich, F., Hesselbach, J.: RCA562: Control Architecture for Parallel Kinematic Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 315–331. Springer, Heidelberg (2010)
38. Kröger, T., Finkemeyer, B., Wahl, F.M.: Manipulation Primitives — A Universal Interface Between Sensor-Based Motion Control and Robot Programming. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 293–313. Springer, Heidelberg (2010)
39. Gough, V.E.: Contribution to discussion of papers on reaearch in automobile stability, control, and tyre performance. In: Proc. of the Institution of Mech. Engineers Auto Division, pp. 392–394 (1956/1957)
40. The MathWorks Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA (2008), `http://www.mathworks.com` (accessed: December 15, 2008)
41. Opal-RT Technologies Inc., 1751 Richardson, Suite 2525, Montréal, Québec, Canada, H3K 1G6 (2008), `http://www.opal-rt.com` (accessed: December 15, 2008)
42. Institut für Robotik und Prozessinformatik at the Technische Universität Carolo-Wilhelmina zu Braunschweig, Mühlenpfordtstr. 23, D-38106 Braunschweig, Germany (2008), `http://www.rob.tu-bs.de/en` (accessed: December 15, 2008)

Manipulation Primitives — A Universal Interface between Sensor-Based Motion Control and Robot Programming

Torsten Kröger, Bernd Finkemeyer, and Friedrich M. Wahl

Abstract. This paper introduces a generic framework for sensor-based robot motion control. The key contribution is the introduction of an adaptive selection matrix for sensor-based hybrid switched-system control. The overall control system consists of multiple sensors and open- and closed-loop controllers, in-between which the adaptive selection matrix can switch discretely in order to supply command variables for low-level controllers of robotic manipulators. How control signals are chosen, is specified by *Manipulation Primitives*, which constitute the interface to higher-level applications. This programming paradigm is formally specified in order to establish the possibility of executing sensor-guided *and* sensor-guarded motion commands simultaneously and in a very open way, such that any kind and any number of sensors can be addressed. A further key feature of this generic approach is, that the control structure can be directly mapped to a corresponding software architecture. The resulting control system is freely scalable depending on the performance requirements of the desired system.

1 Introduction

Serial and parallel kinematic machines are still mainly used for point-to-point operations. Nowadays commercial control units are rarely open for sensor integration.

Torsten Kröger
Stanford University, Artificial Intelligence Laboratory, Department of Computer Science, Stanford, CA 94305-9010, USA
e-mail: tkr@stanford.edu

Bernd Finkemeyer
KUKA Roboter GmbH, Zugspitzstraße 140, 86165 Augsburg, Germany
e-mail: berndfinkemeyer@kuka-roboter.de

Friedrich M. Wahl
Technische Universität Braunschweig, Institute for Robotics and Process Control, Mühlenpfordtstraße 23, 38106 Braunschweig, Germany
e-mail: f.wahl@tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 293–313.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

In research institutions the existing control units are often replaced by own ones in order to perform experiments in control engineering. When looking into the literature, one gets the impression that force/torque control, visual servo control, distance control, manipulator dynamics, and control software architecture issues are solved sufficiently. This is correct for concrete cases. Problems appear when merging several of these fields together. Even if (multi-)sensor integration is provided for a concrete setup for a specific application, many control schemes suffer from bandwidth and switching problems. Engineers developing robotic should be encouraged to have a better view of the *overall* manipulation control system. When considering any kind and any number of sensors (force/torque, distance, vision, pressure, light barriers, etc.), how can even non-experienced programmers implement guarded and guided motion commands w.r.t. any reference frame? What could such a hybrid switched-system control architecture look like? How can sensor signals be consistently mapped to stable, unambiguous, and deterministic manipulator motions? Questions like these are supposed to be answered by the research community to bring existing control approaches into practice.

This contribution addresses exactly this field and suggests a control architecture for any kind of robotic manipulators based on *Manipulation Primitives* (MP). In earlier works, MPs were also denoted as *Skill Primitives* (SP) [1]; both terms are used as synonyms within this document. After related and former works have been presented in Sec. 2, we introduce MPs and present a formal definition in Sec. 3. The following part describes the underlaying control system and the concept of the adaptive selection matrix. For the aim to integrate any number and any kind of sensor into one control system, a high system modularity is required. Such a corresponding software architecture is described in [2, 3, 4, 5]. MPs define the interface between the low-level control system [6], which requires real-time capability, and high-level non-real-time applications [7].

2 Related Work

Within this work, we describe MPs and the underlying control architecture. This section addresses related fields: Classic compliant motion control, which is based on force/pose control concepts; sensor integration methods (e.g., vision and distance sensor integration); robot task specification for sensor-based manipulations; control architectures including software technologies.

Mason presented one of the pioneer works in compliant motion control in 1981 [8]. Based on his work numerous approaches have been published in this field, and especially the group of De Schutter contributed promising concepts to the community (e.g., [9, 10, 11]) and coined the phrase *Task Frame Formalism* (TFF, [6, 12, 13]), which enables the development of compliant motion solutions on an abstract programming level.

These works understand force/torque control as one basic part for compliant motion concepts [14]. As also discussed in [15] and [12], basically three different approaches are known from literature: 1. Impedance control [16], which

uses relationships between acting forces and manipulator poses to adjust the mechanical impedance of the end-effector to external forces. 2. Parallel control [17], which enables to control both, force and pose, along the same task space direction. 3. Force/pose control which controls force and pose in two orthogonal subspaces [18]. The latter reference introduced the term *compliance selection matrix*, which constitutes the basis for the *adaptive selection matrix*. To realize this approach, we have to pay attention to the problem of orthogonality as stated by Duffy [19], who extended the approach, such that it is consistent, independent of units, and independent of any origin coordinate system.

Practical tasks of robot manipulators are usually affected by various uncertainties and inaccuracies. As an obvious result hybrid pose and force/torque control are not always sufficient; the integration of further sensors opens new possibilities and application fields. Especially, the addition of multiple sensors of the same kind, such that the system can situation-dependently benefit from the advantages of single sensors, leads to *robust* system behaviors.

Assuming a robot manipulation system with many different sensors and considering a system that enables the execution of sensor-guided motion control commands in any degree of freedom (DOF), it becomes self-evident that we need to switch discretely between several (open- or closed-loop) continuously working discrete controllers at any time instant. Hence, the analysis of hybrid switched-system control is one fundamental part of the work presented here. Especially, the works of Branicky and Liberzon provide elementary concepts to develop and analyze hybrid switched-system control technologies. The works [20, 21, 22] are considered as the most relevant ones for this paper as will be pointed out in Sec. 4.

Systems clearly become more advanced, when the motion control loop is additionally closed by signals from vision systems. A recent work of Gans and Hutchinson describes, how switched-system control concepts can be used for pose- and image-based visual servo control [23]. In [24] and [25], the integration of visual servo control in the TFF approach is discussed.

Up to now, many research groups published approaches in the field of open control systems allowing the integration of multiple sensors (e.g., [26, 27, 28]). The work by Cortesão et al. [29] presents promising experimental results of force, vision, and pose data fusion.

Control of sensor-based manipulation systems is one major part of this paper, but — in addition — there is still a need for suitable robot programming paradigms allowing this kind of integration and programming when using multiple sensors. One recent and closely related work has been published by De Schutter et al. [30], where a unified constraint-based framework of task specification was presented. For practical implementations, a generic interface is required, which remains unchanged, even when further sensors and/or corresponding controllers are integrated. Besides the parametrization of set-points, such a programming interface must be able to adapt the control system to the current work step and to the required sensor(s). The possibility of expanding the system by any kind of sensor without any expensive rebuilding of the existing (and proven) control system is very important.

This leads to a continuously growing complexity of demands on manipulation control systems, and the development of suitable software control architectures gains in importance as deeply discussed in [2, 3, 4, 5]. A very related and widely known approach is the OROCOS project (Open Robot Control Software, [31]). Another recent suggestion for a distributable and scalable real-time software architecture for software development purposes in robotics was presented by Bäuml et al. in [32]. As also presented in [3], the approach of Maaß et. al. [33] presents a concept for the efficient self-organization of a modular manipulation control system. The most relevant self-references in the context of this paper are [34] and [35].

This contribution extends all known approaches by providing a generic and universal system with *Manipulation Primitives* as top-level interface mapping sensor signals consistently to stable, unambiguous, and deterministic manipulator motions. This universality is extended to the software architecture level, where each software module corresponds to one motion control or sensor module in the overall control scheme.

3 Manipulation Primitives

This section introduces *Manipulation Primitives* (MP) and gives a formal definition, which will be important for the next sections, where the concept of unambiguous mapping of MP parameters to low-level control is explained.

An MP is formally defined as the three-tuple

$$\mathcal{MP} := \{\mathcal{HM}, \tau, \lambda\} \tag{1}$$

where $\mathcal{HM}$ defines a hybrid motion, τ contains tool commands, and the stop condition λ determines the end of execution of a single MP. These three quantities will be described and discussed in the following.

3.1 Hybrid Move $\mathcal{HM}$

$\mathcal{HM}$ defines a hybrid move in the sense of the TFF [9], which becomes extended here in order to take multiple sensors into consideration. In the classical case, a motion command is given w.r.t. the Task Frame and is determined by a six-dimensional vector representing Mason's *Compliance Frame* [8]. This simple vector is not a sufficient and in particular not a practical way to parameterize sensor-guided robot motion commands when considering multiple sensors.

Example 1. A contact transition from free space into any contact state requires a switch from one controller, for example, a feedforward trajectory-following controller, to an adequate other one, for example, a force/torque controller (or vice versa). This switching can only be realized on the low-level control layer (cf. [21] and [22]), i.e., in *real-time* from one control cycle to another. Hence, the responsibility of discretely switching in-between a set of continuously working (discretely realized) controllers cannot be given to the user.

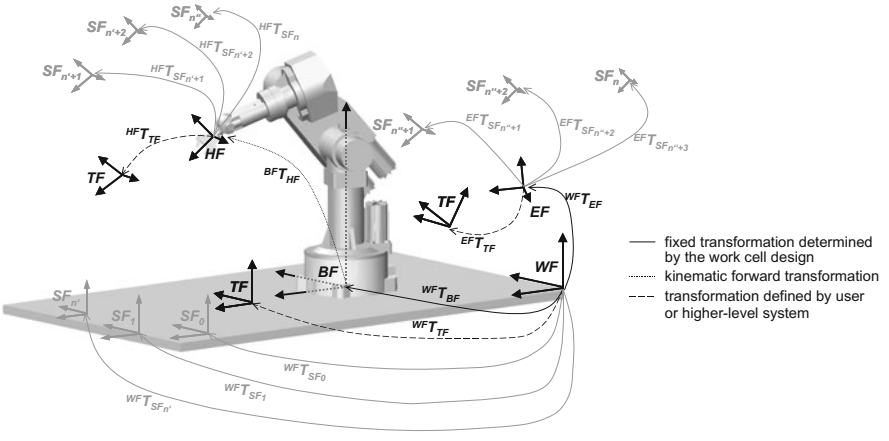

fixed transformation determined by the work cell design
kinematic forward transformation
transformation defined by user or higher-level system

Fig. 1. Frame assignments according to [34] for the proposed TFF specification.

The question of how to specify motion commands unambiguously and universally, such that any sensor can be addressed, must be answered in this context. For this purpose, we define the hybrid motion command $\mathcal{HM}$ as

$$\mathcal{HM} := \{\mathcal{TF}, \mathcal{D}\} \tag{2}$$

with

$$\mathcal{TF} := \{\boldsymbol{\theta}, RF, ANC, FFC\} \tag{3}$$

$$\boldsymbol{\theta} = \left(\theta_x, \theta_y, \theta_z, \theta_{\textcircled{x}}, \theta_{\textcircled{y}}, \theta_{\textcircled{z}}\right)^T \in \mathbb{R}^6 \tag{4}$$

$$RF, ANC \in \{HF, WF, BF, EF\} \tag{5}$$

$$FFC \in \{WF, BF, EF\} \tag{6}$$

and

$$\mathcal{D} := \{D_i | D_i = \iota_i \cup \xi_i;\ \iota_i \in \zeta,\ \forall\, i, j\colon\ i \neq j \implies \iota_i \neq \iota_j;\ i, j \in \{0, \ldots, |\zeta|\}\} \tag{7}$$

with

$$\zeta := \{x, y, z, \textcircled{x}, \textcircled{y}, \textcircled{z}\} \times \{0, \ldots, (m-1)\} \tag{8}$$

$$\iota_i := \{dof\_id_i, level\_id_i\} \in \zeta \tag{9}$$

$$\xi_i := \{value_i, device_i\} \tag{10}$$

Let us discuss eqns. (2) – (10) in the following:

The hybrid move command $\mathcal{HM}$ of eqn. (2) is specified w.r.t. the Task Frame $\mathcal{TF}$, in which a set of set-points $\mathcal{D}$ is applied. The following paragraph explains the Task Frame definition (eqns. (3) – (6)), and subsequently details of $\mathcal{D}$ are introduced (eqns. (7) – (10)).

Definition of the Task Frame w.r.t. Different Reference and Anchor Frames

Fig. 1 shows a non-mobile manipulator with all respective frame assignments in the context of this work. All control values are measured in some Sensor Frame SF_n, which may be located anywhere and must be transformable into the currently applied Task Frame. In contrast to the classical TFF, we admit coupling of the Task Frame to any frame in the work cell. $\boldsymbol{\theta}$ determines the pose of the Task Frame w.r.t. the Reference Frame RF at the beginning of a single MP execution. RF can be selected from a set of frames known by the system (e.g., World Frame (WF), Base Frame (BF), or Hand Frame (HF)). Pose, velocity, and acceleration of each frame is updated every control cycle, typically every millisecond. The frame ANC serves as anchor and connects the Task Frame rigidly to another frame of the work cell. The transformation ${}^{ANC}\mathbf{T}_{TF}$ is constant during the complete MP execution. Hence, the ANC frame has a fundamental influence on the motion to be executed. The frame FFC (eqns. (3) and (6), feedforward compensation) usually equals the World Frame or the robot Base Frame if a mobile manipulation system is considered. If (compliant) motions are to be executed w.r.t. an external moving coordinate system, the FFC frame (i.e., its pose, velocity, and acceleration) enables the internal computation of a feedforward compensation (FFC) signal, such that a sensor-based motion command can be executed in dynamic systems in the same way as in static ones [36]. Regarding the anchor frame, we distinguish between three different cases:

1. Task Frame Attached to the Hand Frame ($ANC = HF$). This case is comparable to the classical TFF [8, 10, 9] and hybrid force/pose control [18]. Here the Task Frame is synonymous to the *Center of Compliance*. It follows the motions of the Hand Frame of the manipulator. All set-points must be interpreted *relatively*

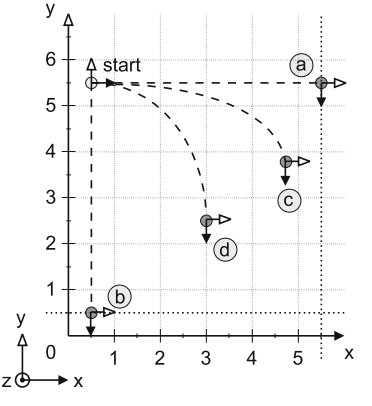


Fig. 2. Task Frame attached to the Hand Frame ($ANC = HF$): Depending on maximum and/or desired angular velocities ω_z^{max} and accelerations α_z^{max} (jerks are assumed to be infinite in this simple example), the set-points $x = 5$, $y = 0$, and $\varphi_z = 90°$ result in different Task Frame poses depicted from a) to d): Only the covered distance is the same in all cases; a) $\alpha_z^{max} \longrightarrow 0$, $\omega_z^{max} \longrightarrow 0$, $a_x^{max} \longrightarrow \infty$, and $v_x^{max} \longrightarrow \infty$; b) $\alpha_z^{max} \longrightarrow \infty$, $\omega_z^{max} \longrightarrow \infty$, $a_x^{max} \longrightarrow 0$, and $v_x^{max} \longrightarrow 0$; c) somewhere between a) and b); d) same as c).

in order to allow unequivocal hybrid switched-system control. Hence, the position and orientation set-points do not determine the target Task Frame pose w.r.t. the old Task Frame, they determine the relative displacements of the Task Frame along the corresponding DOFs. If rotations are involved, the final pose depends on angular velocities, angular accelerations, and angular jerks. For a better understanding, Fig. 2 shows an example with three DOFs.

2. Task Frame Attached to a Fixed Frame ($ANC = WF$, $ANC = BF$)**.** The Task Frame pose is fixed during the execution of MPs. Pose set-points are interpreted in an absolute way and determine the target pose of the hand frame. Classical robot move commands can be realized with this kind of Task Frame configuration. The MP's Task Frame approach implies two major extensions:

(a) As the Task Frame can be placed anywhere within the work cell, the pose set-points for the Hand Frame can be given w.r.t. any frame known by the control system.
(b) Hybrid switched-system control w.r.t. any fixed frame is possible.

The advantages here are: The user is not restricted to one reference frame, and as hybrid switched-system control is available, the programming of sensor-guided and -guarded motions with sensors mounted rigidly in the work cell becomes feasible. It must be kept in mind, that in hybrid switched-system control, pose set-points are fed to an on-line trajectory generator, which acts as stateless feedforward controller [37]. In order to provide hybrid switched-system control w.r.t. fixed frames, pose set-points must be interpreted in a relative way; Fig. 3 illustrates this behavior.

3. Task Frame Attached to an External Frame ($ANC = EF$)**.** The External Frame EF indicates a frame of any device (e.g., a conveyor belt, another manipulator, or a mobile robot). This is similar to the previous case. However, now the Anchor Frame AF is not mandatorily fixed, i.e., the Task Frame may follow a moving frame EF. This clearly simplifies the definition of complex robot tasks specified w.r.t. moving devices by means of a consistent and unique interface. Fig. 4 shows the Task Frame location for a screwing task on a rotating plate. The torque around the z-axis and the force along the z-direction may be controlled w.r.t. the Task Frame

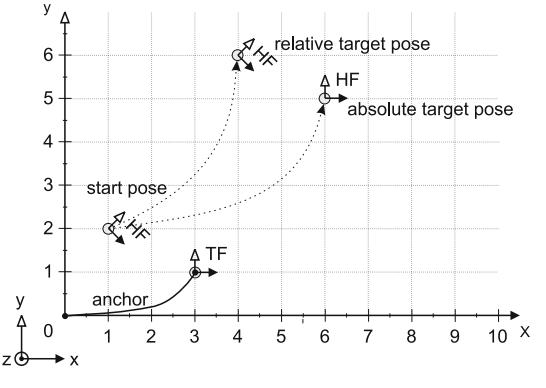


Fig. 3. Task Frame attached to a World Frame ($ANC = WF$): different reached poses of the Hand Frame by interpreting the pose set-points $x = 3$, $y = 4$, and $\varphi_z = 0°$ relatively and absolutely.

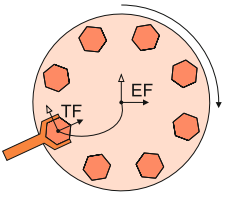


Fig. 4. Robot task: Torque hexagon screws on a rotating plate. The task can be easily defined in the rotating environment EF, and the Task Frame, which is the Reference Frame for all hybrid motions is here anchored with the EF frame.

anchored to the plate. The remaining DOFs may all be pose controlled. As the Task Frame will follow the plate frame, the user must not explicitly consider the rotation ($ANC = EF$), but in this case, the FFC frame has to be specified. Otherwise, the force/torque controller would possess another control behavior w.r.t. the moving EF frame than w.r.t. any static frame (e.g., WF).

Table 1. Tabular example of the MP set-point set $\mathcal{D}_i$. The physical units depend on the selected controller. For each DOF, $(m-1)$ alternatives can be specified to ensure deterministic and stable behavior (cf. [35]).

level_id	x	*device*	y	*device*	z	*device*
0	30	`Pose_Ctrl` in *mm*	-15	`F/T_Ctrl` in N	20	`Distance_Ctrl` in *mm*
1	—	—	0.0	`Distance_Ctrl` in *mm*	15.0	`Velocity_Ctrl` in mm/s
2	—	—	30	`Vision_Ctrl` in *mm*	—	—
...	...	...	...	...	...	...
$m-1$	—	—	—	—	—	—

level_id	ⓧ	*device*	ⓨ	*device*	ⓩ	*device*
0	10	`Vision_Ctrl` in °	1	`F/T_Ctrl` in *Nm*	2	`Pose_Ctrl` in °
1	20	`Pose_Ctrl` in *mm*	10	`Pose_Ctrl` in *mm*	—	—
2	—	—	—	—	—	—
...	...	...	...	...	...	...
$m-1$	—	—	—	—	—	—

Definition of the Set-Point Set $\mathcal{D}$

Assuming, we only apply trajectory-following and force/torque control to a robotic manipulation system with six Cartesian DOFs; the consideration of all possible combinations of pose and force/torque set-points would lead to $2^6 = 64$ hybrid move commands. If controller switchings should be realized with three controllers (e.g., by adding a distance or visual servo controller), the number of motion commands would increase to $3^6 = 729$. Of course, this is not a practical solution. Thus, it has to be possible to determine an alternative set-point individually for each DOF within every control cycle (i.e., on-line). For example, if force/torque control in one direction is not possible, the system should be able to automatically switch to the next alternative (open- or closed-loop) controller for the respective DOF. If the prerequisites to apply this controller are not fulfilled, a further alternative may be chosen etc. If no further alternative can be determined, a save backup controller has to be used. In this way, a deterministic and stable robot behavior is guaranteed in every situation.

Let us now explain eqns. (7) – (10). The set $\mathcal{D}$ contains up to $(6\,m - 1)$ set-points D_i, where m is the number of available controllers; each set-point D_i consists of two attributes:

- ξ_i, a tuple containing $value_i$, which is assigned to a particular controller $device_i$, and
- ι_i defining the DOF dof_id_i, to which the set-point is applied as well as the level $level\_id_i$.

ζ is a set of possible DOF and level combinations and generally contains $|\zeta| = 6\,m$ elements. Table 1 presents an example of a set-point set $\mathcal{D}$. It demonstrates that a list of alternative devices may be declared for each DOF. Of course, these alternatives are optional, and its number depends on the current task. They are indicated unambiguously by the level ID value ($level\_id_i \in \mathbb{N}$). A set-point with $level\_id = 0$ represents a desired robot state of first choice. $level\_id = 1$ denotes the first alternative set-point and so on. For example, in y-direction force control with a desired value of $(-15\,N)$ has been chosen. If force control is not possible (e.g., because there is no contact), the corresponding DOF will automatically be controlled by a distance controller with a set-point of 0 *mm*. If this controller is also not able to service the DOF (e.g., in case the robot is outside of the sensor's measurement range), the determined open-loop velocity controller is activated. If no further alternative is available, a save default backup controller will be activated. In general, a pose controller will keep the robot in a stable state in this situation and the current MP would be terminated with an error state. Of course, any combination of controllers and set-points is possible as a matter of principle. By means of the attribute $device_i$, it is possible to choose the optimal controller for each task, i.e., the best appropriate force or visual servo control concept etc. comes to execution. The necessary switching processes are handled by the *adaptive selection matrix*, which will be outlined in Sec. 4.

3.2 Tool Command τ

The second part of an MP, the *tool command* τ, allows the integration and the access to any tool (gripper, drilling machine, welding apparatus, etc.) mounted to the manipulator's end-effector or to any device of the work cell. It is possible to start the execution of one or more tools synchronously with one MP. The tool command formally is defined as

$$\tau := \{\tau_i | \tau_i = \{tool\_name, command\}\} \tag{11}$$

The structure of the *command* attribute is kept very open and very general in order to permit the integration of very complex tools.

3.3 Stop Condition λ

The *stop condition* defines the termination of an MP. A single MP runs until the sensor states and the robot state, which are defined by the user in the stop condition λ, are reached. This is an important point: In robot programming paradigms widely used up to now, the stop of motion is implicitly defined by the *move* command. In the MP programming paradigm, the end of a single motion command ($\mathcal{HM}$) is separated from the motion description. This leads to a higher flexibility regarding the simultaneous execution of sensor-guided *and* sensor-guarded motions, which even may exhibit fault tolerances due to kinematic inaccuracies and to uncertainties in the environment.

We distinguish between default and user defined stop conditions. The default stop condition incorporates possible system errors, for example, leaving the workspace or exceeding sensor signals, for example, due to too high contact forces. It leads to an interruption of the current MP execution. The default stop condition depends on the involved control devices and the used tool devices; it is adapted automatically.

The user stop condition is a Boolean expression defined as

$$\lambda := \mathcal{S} \longrightarrow \{true, false\} \tag{12}$$

where $\mathcal{S}$ is the set of available sensors and their corresponding filter functions.

$$\begin{aligned} \lambda_u = & \left(\overline{{}^{task}F_z}^{20\,ms} < -15\,N\right) \vee \left(\left({}^{world}p_x^{hand} > 40\,mm\right)\right. \\ & \left. \wedge \left(|{}^{world}v_x^{task}| > 75\,mm/s\right)\right) \vee \left(t \geq 5000\,ms\right) \end{aligned} \tag{13}$$

Eqn. (13) gives an example of a user-defined stop condition that would let the execution of an MP end if the force of the Task Frame's z-direction averaged over 20 milliseconds gets below -15 N, after a timeout of five seconds, or if the origin of the manipulator's Hand Frame w.r.t. the World Frame gets above 40 mm and the absolute velocity of the Task Frame's origin w.r.t. the World Frame's x-axis gets higher than 75 mm/s.

4 Control System

This section gives an outline of a control system to execute MPs. As the involved variables and controllers may vary from DOF to DOF and from MP to MP, the control structure has to be adapted instantaneously according to any given situation.

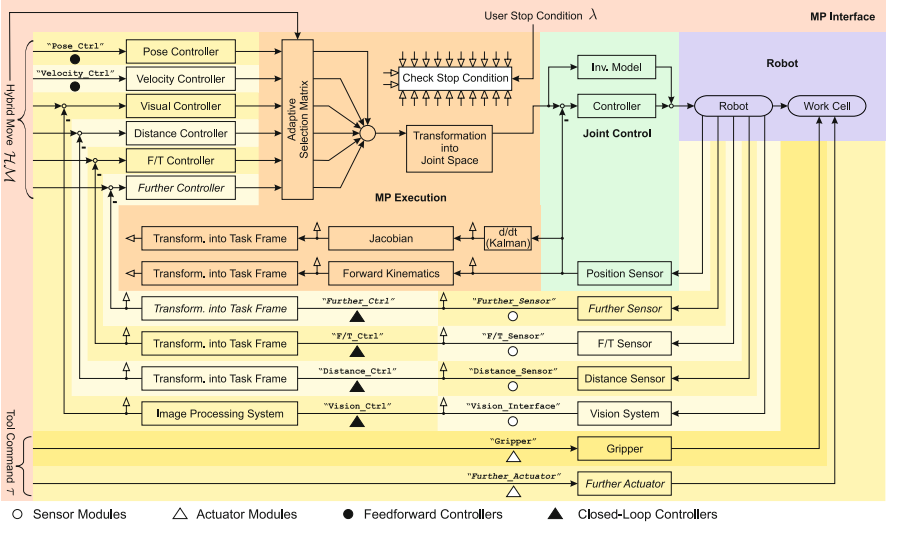


Fig. 5. Overview of the hybrid switched-system control scheme to execute Manipulation Primitives.

4.1 Control Architecture

The overall control architecture can be considered a hybrid switched-system [21, 22]. Fig. 5 depicts a rough overview of such a control scheme. Here, hybrid switched-system control of pose, velocity, force/torque (F/T), distance, and vision is indicated. Of course, this set can be expanded by any other physical variable and/or controller, respectively. At first glance, the structure seems similar to common hybrid switched-system control structures. But in contrast to common hybrid switched-system controllers, the selection matrix is not static during one robot command: It considers the current robot and environment state. Thus, it is an *adaptive selection matrix*.

4.2 The Adaptive Selection Matrix

As described in Sec. 3, the MP approach allows the definition of several alternative set-points (eqns. (7) – (10)), such that a stable and efficient robot control is feasible for each individual DOF.

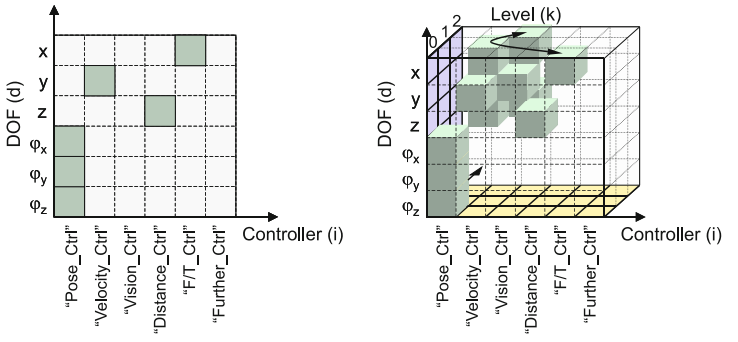


Fig. 6. Classical two-dimensional assignment (left) and adaptive three-dimensional assignment (right) of controllers and DOFs.

In common hybrid switched-system control approaches, selection matrices $\mathbf{S}_i$ are generated from the compliance frame for each available controller type. The corresponding controller is addressed by the index i. The six-dimensional resulting control variable vector $\boldsymbol{o}$ is computed out of the n available control variable vectors $\boldsymbol{o_i}$ of the involved controllers in the following way:

$$\boldsymbol{o} \;=\; \sum_{i=0}^{n-1} \mathbf{S}_i \, \boldsymbol{o}_i \tag{14}$$

where the following condition with the identity matrix $\mathbf{I}$ must be fulfilled

$$\sum_{i=0}^{n-1} \mathbf{S}_i \;=\; \mathbf{I} \tag{15}$$

This results in the required unique assignment of controllers to DOFs. Fig. 6 (left) depicts a sample assignment.

For the execution of MPs, which allow the usage of any kind and any number of sensors, this simple two-dimensional view of selection matrices does not suffice. The definition of several alternative control loops requires the extension by a third dimension, representing the *control level*. This three-dimensional perception is illustrated in the right part of Fig. 6. The control loops of a DOF d are shifted along the double-headed arrow. Thus, the active control variable can no longer be determined by constant selection matrices $\mathbf{S}_i$ of eqn. (14). The selection works dynamically and depends on two factors:

(a) The currently available sensors and controllers as well as the current system state.
(b) The assignment of controllers and control levels per DOF.

Formally, a controller i delivers three matrices: the assignment matrix $\mathbf{Z}_i$, the control matrix $\mathbf{O}_i$, and the availability matrix $\mathbf{F}_i$. They are the basis for the calculation of the control variable assignment matrix $\mathbf{E}_o$ and flag assignment matrix $\mathbf{E}_f$, which are used to determine the selection matrix $\mathbf{L}$, from which the resulting control value

vector $\boldsymbol{o}$ can be derived. All these matrices and all calculation steps will be explained in the following.

Assignment Matrices $\mathbf{Z}_i$. As described in Sec. 3, it is possible to determine alternatives for each set-point. Thus, each DOF d contains several control levels k. The assignment of the i-th controller to a control level k is formally represented by an assignment matrix $\mathbf{Z}_i$:

$$\mathbf{Z}_i = \begin{pmatrix} z_{0,x} & z_{0,y} & \cdots & z_{0,\textcircled{z}} \\ z_{1,x} & z_{1,y} & \cdots & z_{1,\textcircled{z}} \\ \vdots & \vdots & \vdots & \vdots \\ z_{k,x} & z_{k,y} & \cdots & z_{k,\textcircled{z}} \\ \vdots & \vdots & \vdots & \vdots \\ z_{(m-1),x} & z_{(m-1),y} & \cdots & z_{(m-1),\textcircled{z}} \end{pmatrix} \in \mathbb{B}^{m\times 6} \text{ , where } \mathbb{B} = \{0, 1\} \tag{16}$$

Each column corresponds to a DOF and each row represents a control level, where m determines the maximum number of control levels. An entry of '1' assigns the DOF of the i-th controller to the control level k. The matrix corresponds to a vertical slice of the three-dimensional adaptive selection matrix representation of Fig. 6 (right).

Example 2.

$$\mathbf{Z}_{\texttt{Velocity\_Ctrl}} = \begin{pmatrix} 0\,1\,0\,0\,0\,0 \\ 1\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 \end{pmatrix} \tag{17}$$

Here, the controller is selected to manage the y-direction on level 0 and to manage the x- and z-direction on level 1. All remaining DOFs are controlled by some other controller.

The assignment matrices $\mathbf{Z}_i$ are implicitly defined by the set-point set $\mathcal{D}$ and are automatically computed by the control system.

Control Matrices $\mathbf{O}_i$. The control variables of one single controller i are summarized as vector $\boldsymbol{o}_i$. The dimension of the vector corresponds to the number of available DOFs of the robotic system. For computing the adaptive selection matrix, the control variables are represented by the diagonal elements of a control matrix $\mathbf{O}_i$. For a robot with six DOFs, the matrix $\mathbf{O}_i$ for one single device i is defined as

$$\mathbf{O}_i = \begin{pmatrix} o_x & 0 & 0 & 0 & 0 & 0 \\ 0 & o_y & 0 & 0 & 0 & 0 \\ 0 & 0 & o_z & 0 & 0 & 0 \\ 0 & 0 & 0 & o_{\textcircled{x}} & 0 & 0 \\ 0 & 0 & 0 & 0 & o_{\textcircled{y}} & 0 \\ 0 & 0 & 0 & 0 & 0 & o_{\textcircled{z}} \end{pmatrix} \in \mathbb{R}^{6\times 6} \tag{18}$$

Thus, the common control variable vector $\boldsymbol{o}_i$ of the i-th controller can be derived from $\mathbf{O}_i$ as follows:

$$\boldsymbol{o}_i = \operatorname{diag}(\mathbf{O}_i) \tag{19}$$

Availability Matrix $\mathbf{F}_i$. To avoid the usage of control variables of inoperative control loops (e.g., force/torque control in free space), the validity is indicated by a flag. A value of '1' denotes a usable control variable and '0' denotes an invalid control variable. The elements of the availability matrix $\mathbf{F}_i$ contain the mentioned flags of the i-th controller. For six DOFs, $\mathbf{F}_i$ is given by

$$\mathbf{F}_i = \begin{pmatrix} f_x & 0 & 0 & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 & 0 & 0 \\ 0 & 0 & f_z & 0 & 0 & 0 \\ 0 & 0 & 0 & f_{\textcircled{x}} & 0 & 0 \\ 0 & 0 & 0 & 0 & f_{\textcircled{y}} & 0 \\ 0 & 0 & 0 & 0 & 0 & f_{\textcircled{z}} \end{pmatrix} \in \mathbb{B}^{6\times 6} \tag{20}$$

The availability flag vector $\boldsymbol{f}_i$ of the i-th controller can be written as

$$\boldsymbol{f}_i = \operatorname{diag}(\mathbf{F}_i) \tag{21}$$

Control Variable Assignment Matrix $\mathbf{E}_o$. The calculation of the resulting control value vector $\boldsymbol{o}$ requires the mapping of the control values of all controllers to the corresponding DOF *d and* control level k. This is possible, as the definition of eqn. (7) ensures, that one and only one controller can be chosen per DOF and level. The mapping is represented by the control variable assignment matrix $\mathbf{E}_o$. The calculation of $\mathbf{E}_o$ with n involved controllers leads to

$$\mathbf{E}_o = \sum_{i=0}^{n-1} \mathbf{Z}_i \, \mathbf{O}_i = \begin{pmatrix} o'_{0,x} & o'_{0,y} & \cdots & o'_{0,\textcircled{z}} \\ o'_{1,x} & o'_{1,y} & \cdots & o'_{1,\textcircled{z}} \\ \vdots & \vdots & \vdots & \vdots \\ o'_{k,x} & o'_{k,y} & \cdots & o'_{k,\textcircled{z}} \\ \vdots & \vdots & \vdots & \vdots \\ o'_{(m-1),x} & o'_{(m-1),y} & \cdots & o'_{(m-1),\textcircled{z}} \end{pmatrix} \in \mathbb{R}^{m\times 6} \tag{22}$$

Each column corresponds to a DOF d and each row to one particular control level k.

Selection Matrix L. To calculate the control value vector $\boldsymbol{o}$, which contains the input values for the *joint controller* (cf. Fig. 5), the correct row of $\mathbf{E}_o$ must be chosen for each single DOF. This is realized by the selection matrix $\mathbf{L}$. It is defined as

$$\mathbf{L} = \begin{pmatrix} l_{0,x} & l_{0,y} & \cdots & l_{0,\textcircled{z}} \\ l_{1,x} & l_{1,y} & \cdots & l_{1,\textcircled{z}} \\ \vdots & \vdots & \vdots & \vdots \\ l_{k,x} & l_{k,y} & \cdots & l_{k,\textcircled{z}} \\ \vdots & \vdots & \vdots & \vdots \\ l_{(m-1),x} & l_{(m-1),y} & \cdots & l_{(m-1),\textcircled{z}} \end{pmatrix} \in \mathbb{B}^{m\times 6} \tag{23}$$

The columns are assigned to the system DOFs, and the rows are assigned to control levels. The entries contain '1' or '0', where '1' selects the corresponding value. Of course, it exists exactly one '1' per column. Thus, we obtain

$$\sum_{k=0}^{m-1} l_{k,d} \stackrel{!}{=} 1 \text{ with } d \in \left\{x, y, z, \textcircled{x}, \textcircled{y}, \textcircled{z}\right\} \tag{24}$$

Transposing $\mathbf{L}$ and multiplying it with $\mathbf{E}_o$ results in a symmetric matrix, whose diagonal elements contain the resulting control value vector $\boldsymbol{o}$ (cf. eqn. (14)). It can be written as

$$\boldsymbol{o} = \operatorname{diag}\left(\mathbf{L}^T \mathbf{E}_o\right) \tag{25}$$

Please note that the elements of $\mathbf{L}$ are still undetermined. For its calculation the flag assignment matrix $\mathbf{E}_f$ is responsible.

Flag Assignment Matrix $\mathbf{E}_f$. The matrix $\mathbf{L}$ depends on the n available $\mathbf{F}_i$ matrices of the n involved controllers (with $i \in \{0, \ldots, n-1\}$ and $n \leq 6\,m$) and on the assignment matrixes $\mathbf{Z}_i \; \forall \; i \in \{0, \ldots, n-1\}$. This results in the flag assignment matrix

$$\mathbf{E}_f = \sum_{i=0}^{n-1} \mathbf{Z}_i \, \mathbf{F}_i = \begin{pmatrix} f'_{0,x} & f'_{0,y} & \cdots & f'_{0,\textcircled{z}} \\ f'_{1,x} & f'_{1,y} & \cdots & f'_{1,\textcircled{z}} \\ \vdots & \vdots & \vdots & \vdots \\ f'_{k,x} & f'_{k,y} & \cdots & f'_{k,\textcircled{z}} \\ \vdots & \vdots & \vdots & \vdots \\ f'_{(m-1),x} & f'_{(m-1),y} & \cdots & f'_{(m-1),\textcircled{z}} \end{pmatrix} \in \mathbb{B}^{m \times 6} \tag{26}$$

4.2.1 Determination of L and *o*

The columns (DOFs) of $\mathbf{E}_f$ can be mapped to the corresponding columns of $\mathbf{L}$. The mapping law for each single DOF d ($d \in \{x, y, z, \textcircled{x}, \textcircled{y}, \textcircled{z}\}$) is given in the following table. A '×' entry means *'don't care'*.

$f'_{0,d}$	$f'_{1,d}$	...	$f'_{k,d}$	...	$f'_{(m-1),d}$	$l_{0,d}$	$l_{1,d}$	...	$l_{k,d}$	...	$l_{(m-1),d}$
0	0	...	0	...	0	0	0	...	0	...	0
1	×	...	×	...	×	1	0	...	0	...	0
0	1	...	×	...	×	0	1	...	0	...	0
0	0	...	1	...	×	0	0	...	1	...	0
0	0	...	0	...	1	0	0	...	0	...	1

As can be seen from the table, one and only one controller is active per DOF, such that it is in accordance with eqn. (24). The available controller with the lowest *level_id* will be activated. The first line of the table indicates an error state: No controller of the current MP is able to service the DOF. In such a case, the MP will be terminated, and the corresponding DOF will be controlled by a backup controller. The table can be rewritten as

$$
\begin{aligned}
l_{0,d} &= f'_{0,d} \\
l_{1,d} &= \overline{f'_{0,d}} \wedge f'_{1,d} \\
&\vdots \\
l_{k,d} &= \overline{f'_{0,d}} \wedge \overline{f'_{1,d}} \wedge \cdots \wedge f'_{k,d} \\
&\vdots \\
l_{(m-1),d} &= \overline{f'_{0,d}} \wedge \overline{f'_{1,d}} \wedge \cdots \wedge \overline{f'_{k,d}} \wedge \cdots \wedge f'_{(m-1),d}
\end{aligned}
\tag{27}
$$

As these terms become very complex for multiple levels, we introduce the Boolean variable $\rho_{k,d}$

$$
\rho_{k,d} := \overline{f'_{k,d}} \wedge \rho_{(k-1),d}, \text{ where } k \in \{0, \ldots, (m-1)\} \text{ and } \rho_{(-1),d} := 1 \tag{28}
$$

Now eqn. (27) can be rewritten as

$$
\begin{aligned}
l_{0,d} &= f'_{0,d} \\
l_{1,d} &= \rho_{0,d} \wedge f'_{1,d} \\
&\vdots \\
l_{k,d} &= \rho_{(k-1),d} \wedge f'_{k,d} \\
&\vdots \\
l_{(m-1),d} &= \rho_{(m-2),d} \wedge f'_{(m-1),d}
\end{aligned}
\tag{29}
$$

By means of this recursive mapping rule and by applying eqn. (25), $\boldsymbol{o}$ can be computed as

$$
\boldsymbol{o} = \begin{pmatrix}
\sum_{k=0}^{m-1} & o'_{k,x} & (\rho_{(k-1),x} & \wedge & f'_{k,x}) \\
\sum_{k=0}^{m-1} & o'_{k,y} & (\rho_{(k-1),y} & \wedge & f'_{k,y}) \\
\sum_{k=0}^{m-1} & o'_{k,z} & (\rho_{(k-1),z} & \wedge & f'_{k,z}) \\
\sum_{k=0}^{m-1} & o'_{k,\textcircled{x}} & (\rho_{(k-1),\textcircled{x}} & \wedge & f'_{k,\textcircled{x}}) \\
\sum_{k=0}^{m-1} & o'_{k,\textcircled{y}} & (\rho_{(k-1),\textcircled{y}} & \wedge & f'_{k,\textcircled{y}}) \\
\sum_{k=0}^{m-1} & o'_{k,\textcircled{z}} & (\rho_{(k-1),\textcircled{z}} & \wedge & f'_{k,\textcircled{z}})
\end{pmatrix}
\tag{30}
$$

Fig. 7 presents the resulting block diagram of eqn. (30) for one DOF. It summarizes the mathematical representation of the *adaptive selection matrix* from a technical point of view.

4.3 Final Remarks on the Availability Flag Vector $\boldsymbol{f_i}$

The availability flag vector decides if a control loop is currently able to cope with the current system state. Thus, each element of $\boldsymbol{f}_i$ is determined by a function that maps

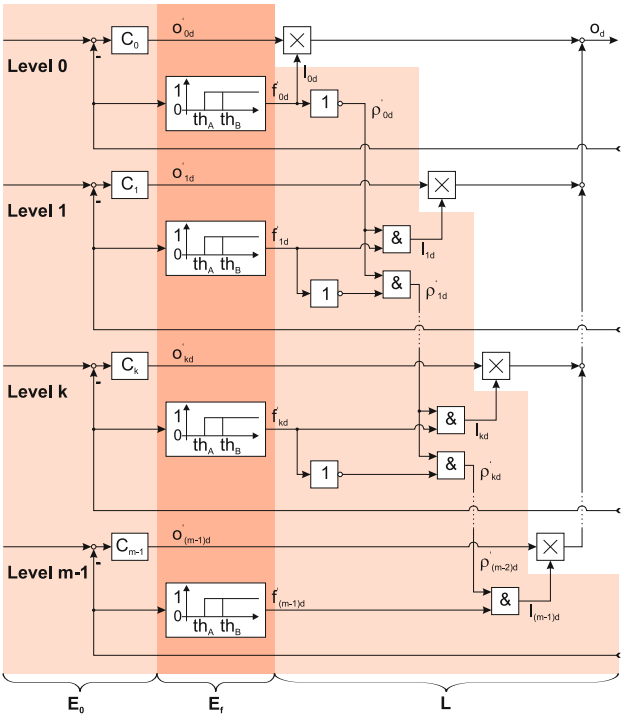


Fig. 7. Technical representation for one DOF of the adaptive selection matrix.

state variables, sensor signals, or any other events to a logical value. One element d of $\boldsymbol{f}_i$ of the i-th controller can be written as

$$f_{i,d} := \mathcal{S} \longrightarrow \mathbb{B} \tag{31}$$

where $\mathcal{S}$ is the set of available sensor signals (cf. eqn. 12). The mapping function depends on the corresponding controller. It can be a simple constant ($f_{i,d} = 1$), the result of a comparison with a threshold (th_A and th_B, cf. Fig. 7), or any complex function. By means of this flag function stability of each continuously working discrete controller in the hybrid switched-system must be guaranteed.

In this way, many critical situations can be handled. Some examples are outlined in the following:

- Managing the transition from free space motions to force/torque controlled or any sensor-guided compliant motions, respectively.
- Any internal controller errors, for example, a missing sensor signal, can set the flag to zero.
- If switching to a controller yields heavy jerk values in the robot's end-effector or drives, it can be avoided by this flag.
- It becomes possible to implement monitoring and safety functions with the adaptive selection matrix.

As already stated in Sec. 2, the problem of orthogonality is very relevant. Compared to classic approaches [8, 10, 11], orthogonality is always guaranteed by the adaptive selection matrix in principle.

This section introduced the adaptive selection matrix — the main contribution of this paper. One further important issue concerns the switching behavior of the adaptive selection matrix. Predefined trajectories as they are common in the states of research and technology are unsuited. All non-sensor-guided motions must be generated by an on-line trajectory generator, which is able to take over control for one or more DOFs from any arbitrary system state [37].

5 Practical Results

In order to save space, we refer to other publications showing experimental as well as practical results achieved with this generic concept. [7, 6, 12] show concrete results that were achieved with this generically introduced framework. Furthermore, the manipulator of [38] plays the parlor game Jenga [39], and applies force/torque, acceleration, vision, and distance sensors in one exhibit. [40] and [41] deliver further details of this concept. How the concept of MPs can be used as an interface to higher-level motion planning systems, which consider inaccuracies in the kinematics as well as in the environment (e.g., [30]), can be found in [34].

6 Conclusion

We introduced a very basic but nevertheless powerful and generic interface represented by *Manipulation Primitives* to simultaneously specify sensor-guided and sensor-guarded robot motion commands, which are executed by a hybrid switched-system controller. As a consequence (multi-)sensor integration in robotic manipulation control systems becomes strongly simplified. The introduced programming paradigm constitutes an intermediate interface above low-level control layers including hybrid switched-system control.

We would like to encourage developers to reason on practical implementations of the Task Frame Formalism in the sense proposed in this contribution. As a result, the potential of this formalism becomes clearer and thus enables the robotics research community to provide a further milestone to industrial automation.

Acknowledgements. The authors are deeply indebted to the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), whose funding of the Sonderforschungsbereich 562 (SFB, Collaborative Research Center 562) made this work possible. All the fruitful discussions with Michael Kolbus, Jochen Maaß, Thomas Reisinger, and Ulrike Thomas are highly appreciated. Furthermore, the authors would like to thank *QNX Software Systems* for providing free software licenses.

References

1. Mosemann, H., Wahl, F.M.: Automatic decomposition of planned assembly sequences into skill primitives. IEEE Trans. on Robotics and Automation 17(5), 709–718 (2001)
2. Finkemeyer, B., Kröger, T., Wahl, F.M.: A Middleware for High-Speed Distributed Real-Time Communication. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 193–212. Springer, Heidelberg (2010)
3. Steiner, J., Maaß, J., Goltz, U.: Self-Management within a Software Architecture for Parallel Kinematic Machines. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 355–371. Springer, Heidelberg (2010)
4. Dadji, Y., Michalik, H., Kohn, P.N., Steiner, J., Beckmann, G., Möglich, T., Varchmin, J.-U.: A Communication Architecture for Distributed Real-Time Robot Control. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 213–231. Springer, Heidelberg (2010)
5. Steiner, J., Diethers, K., Goltz, U.: Model Based Quality Assurance for a Robotic Software Architecture. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 373–389. Springer, Heidelberg (2010)
6. Maaß, J., Dietrich, F., Hesselbach, J.: RCA562: Control Architecture for Parallel Kinematic Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 315–331. Springer, Heidelberg (2010)
7. Thomas, U., Wahl, F.M.: Assembly Planning and Task Planning–Two Prerequisites for Automated Robot Programming. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 333–354. Springer, Heidelberg (2010)
8. Mason, M.T.: Compliance and force control for computer controlled manipulators. IEEE Trans. on Systems, Man, and Cybernetics 11, 418–432 (1981)
9. Bruyninckx, H., Schutter, J.D.: Specification of force-controlled actions in the task frame formalism — A synthesis. IEEE Trans. on Robotics and Automation 12(4), 581–589 (1996)
10. Schutter, J., van Brussel, J.: Compliant robot motion I. A formalism for specifying compliant motion tasks. The International Journal of Robotics Research 7(5), 3–17 (1988)
11. Schutter, J., van Brussel, J.: Compliant robot motion II. A control approach based on external control loops. The International Journal of Robotics Research 7(4), 18–33 (1988)
12. Reisinger, T., Kolbus, M.,Wobbe, F., Schumacher, W.: Integrated Force and Motion Control of Parallel Robots – Part 2: Constrained Space. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 253–273. Springer, Heidelberg (2010)
13. Kröger, T., Finkemeyer, B., Wahl, F.M.: A task frame formalism for practical implementations. In: Proc. of the IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, pp. 5218–5223 (2004)
14. Villani, L., Schutter, J.D.: Force control. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics, 1st edn., ch. 7, pp. 161–185. Springer, Heidelberg (2008)
15. Osypiuk, R., Kröger, T.: Parallel Stiffness Actuators with Six Degrees of Freedom for Efficient Force/Torque Control Applications. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 275–291. Springer, Heidelberg (2010)
16. Hogan, N.: Impedance control: An approach to manipulation. Part I: Theory. Part II: Implementation. Part III: Applications. ASME Journal of Dynamic Systems, Measurment, and Control 107, 1–24 (1985)
17. Chiaverini, S., Sciavicco, L.: The parallel approach to force/position control of robotic manipulators. IEEE Trans. on Robotics and Automation 9(4), 361–373 (1993)

18. Raibert, M.H., Craig, J.J.: Hybrid position/force control of manipulators. ASME Journal of Dynamic Systems, Measurement and Control 102, 126–133 (1981)
19. Duffy, J.: The fallacy of modern hybrid control theory that is based on "orthogonal complements" of twist and wrench spaces. Journal of Robotic Systems 7(2), 139–144 (1990)
20. Branicky, M.S.: Studies in Hybrid Systems: Modeling, Analysis, and Control. Ph.D. thesis, Electrical Engineering and Computer Science Dept., Massachusetts Institute of Technology (1995), http://dora.cwru.edu/msb/pubs.html (accessed: December 15, 2008)
21. Branicky, M.S.: Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. IEEE Trans. on Automatic Control 43(4), 475–482 (1998)
22. Liberzon, D.: Switching in Systems and Control. In: Systems and Control: Foundations and Applications, Birkhäuser, Boston (2003)
23. Gans, N.R., Hutchinson, S.A.: Stable visual servoing through hybrid switched-system control. IEEE Trans. on Robotics 23(3), 530–540 (2007)
24. Baeten, J., Schutter, J.D.: Integrated Visual Servoing and Force Control. Springer Tracts in Advanced Robotics, vol. 8. Springer, Heidelberg (2004)
25. Chaumentte, F., Hutchinson, S.A.: Visual servoing and visual tracking. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics, 1st edn., ch. 24, pp. 563–583. Springer, Heidelberg (2008)
26. Borelly, J.-J., Coste-Maniere, E., Espiau, B., Kapellos, K., Pissard-Gibollet, R., Simon, D., Turro, N.: The orccad architecture. The International Journal of Robotics Research 17(4), 338–359 (1998)
27. Bruyninckx, H., Soetens, P., Koninckx, B.: The real-time motion core of the orocos project. In: Proc. of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, pp. 2766–2771 (2003)
28. Scheider, S.A., Chen, V.W., Pardo-Castellote, G., Wang, H.H.: Controlshell: A software architecture for complex electromechanical systems. The International Journal of Robotics Research 17(4), 360–380 (1998)
29. Cortesão, R., Koeppe, R., Hirzinger, G.: Data fusion for robotic assembly tasks based on human skills. IEEE Trans. on Robotics 20(6), 941–952 (2004)
30. Schutter, J., Laet, T.D., Rutgeerts, J., Decré, W., Smits, R., Aertbeliën, E., Claes, K., Bruyninckx, H.: Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. The International Journal of Robotics Research 26(5), 433–454 (2007)
31. OROCOS Homepage. Open robot control software (2002), http://www.orocos.org (accessed: December 15, 2008)
32. Bäuml, B., Hirzinger, G.: Agile robot development (aRD): A pragmatic approach to robotic software. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, pp. 3741–3748 (2006)
33. Maaß, J., Steiner, J., Raatz, A., Hesselbach, J., Goltz, U., Amado, A.: Self-management in a control architecture for parallel kinematic robots. In: Proc. of the 27th ASME Computers and Information in Engineering Conference, New York, NY, USA (2008)
34. Finkemeyer, B., Kröger, T., Wahl, F.M.: Executing assembly tasks specified by manipulation primitive nets. Advanced Robotics 19(5), 591–611 (2005)
35. Finkemeyer, B.: Robotersteuerungsarchitektur auf der Basis von Aktionsprimitiven (in German). Shaker Verlag, Aachen (2004)
36. Thomas, U., Wahl, F.M., Maaß, J., Hesselbach, J.: Towards a new concept of robot programming in high speed assembly applications. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Canada, pp. 3827–3833 (2005)

37. Kröger, T.: On-Line Trajectory Generation in Robotic Systems. Ph.D. thesis, Institut für Robotik und Prozessinformatik, Technische Universität Carolo-Wilhelmina zu Braunschweig (2009)
38. Kröger, T., Finkemeyer, B., Winkelbach, S., Molkenstruck, S., Eble, L.-O., Wahl, F.M.: A manipulator plays Jenga. IEEE Robotics and Automation Magazine 15(3), 79–84 (2008)
39. Hasbro Inc., 1027 Newport Avenue, Mailstop A906, Pawtucket, RI 02861, USA. Jenga homepage (2008), `http://www.jenga.com` (accessed: December 15, 2008)
40. Kröger, T., Finkemeyer, B., Winkelbach, S., Molkenstruck, S., Eble, L.-O., Wahl, F.M.: Demonstration of multi-sensor integration in industrial manipulation (poster). In: Proc. of the IEEE International Conference on Robotics and Automation, Orlando, FL, USA, pp. 4282–4284 (2006)
41. Kröger, T., Finkemeyer, B., Winkelbach, S., Molkenstruck, S., Eble, L.-O., Wahl, F.M.: Demonstration of multi-sensor integration in industrial manipulation (video). In: Proc. of the IEEE International Conference on Robotics and Automation, Orlando, FL, USA (2006)

RCA562: Control Architecture for Parallel Kinematic Robots

Franz Dietrich, Jochen Maaß, Annika Raatz, and Jürgen Hesselbach

Abstract. The design of powerful control that suits multiple types of parallel kinematic robots is extraordinarily challenging. The diversity of parallel kinematics and their optimization that customizes them to specific tasks require highly individualized control functionalities. This work intends to provide principles, methods and tools for the development of such control software. It aims at the time-efficient realization of custom robot controllers that suit particular application domains and use hardware resources as sophisticated as possible. A task-frame formalism for trajectory generation is defined exploiting the full potential of parallel robots. This formalism can be used as a generic programming interface for parallel robots. Design patterns for so-called active connectors, modular motion planning, sensor integration and restricted state machines in token-passing context are discussed with regard to control of parallel robots. *RCA562*, a control application for parallel robots incorporating this knowledge, serves as an illustrative validation example.

1 Introduction and Motivation

The structure and the geometry of parallel robots can be optimized effectively to suit specific applications [1]. For this reason the variety of structures and the task specialization are higher compared to serial kinematic robots. In consequence the implementation of a powerful control for parallel kinematic robots is extraordinarily challenging. Both, the demand for custom functionalities of a specific robot and the effort spent on the realization must be considered. This contribution provides an

Franz Dietrich · Annika Raatz · Jürgen Hesselbach
Technische Universität Braunschweig, Institute of Production Automation and Machine Tools, Langer Kamp 19b, 38106 Braunschweig, Germany
e-mail: {f.dietrich,a.raatz,j.hesselbach}@tu-bs.de

Jochen Maaß
Skysails GmbH & Co. KG, Veritaskai 3, 21079 Hamburg, Germany
e-mail: j.maass@tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 315–331.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

overview on methods for straight-forward and time-efficient realization of a custom robot controller which is specialized for the application domain demanded.

In section 2 a modularizing formalism for motion generation is developed, forming the basis for highly flexible robot control. It emphasises the aim of creating a high proportion of reusable components. This formalism makes use of the concept of task-oriented robot programming. In section 3 a new design methodology based on this formalism is elaborated, taking the high level of model integration in control and signal processing into account. This method applies the techniques of model-based software development. In this way reuseable solution patterns are obtained. These are intended to solve the specific and recurring design problems in the domain of the realization of control functionalities for parallel kinematic robots.

2 Definition of a Formalism for Trajectory Generation

The task frame formalism was introduced by Mason [2] and refined by deSchutter [3]. Hybrid control is a core element of this concept, i.e. sensor integration for the control of selected degrees of freedom, instead of taught-in positions exclusively. This formalism allows assembly sequences to be realized in an error tolerant way. Hasegawa et al. [4] proposed robot programming based on skill primitives and presented an implentation of hybrid control for contact situations. This publication and the contributions regarding programming and execution of automated assembly of the research group of Wahl (e.g. [5, 6, 7]) inspired this work. When parallel robots are used for assembly tasks, the repeatability increases significantly. In addition parallel robots are capable of performing better than serial ones regarding compliant motion due to higher stiffness and lower inertia.

Similar to the task frame implementations presented in [8] a task frame formalism is defined which allows the full potential of parallel robots to be used. At first, a primitive is defined (for explanation of the symbols, see the text below):

$$\mathcal{MP} := \left\{ \begin{array}{ll} \mathsf{RF} & \text{(Reference Frame)} \\ \mathcal{M} & \text{(Movement Description)} \\ \mathcal{T} & \text{(Transition Condition)} \\ \mathcal{R} = \biguplus_i \mathsf{SR}_i & \text{(Return Value)} \end{array} \right\}. \tag{1}$$

Since models of advanced control schemes for parallel kinematics are usually formulated in operational space [9], it is recommendable to define the trajectory generation TG in state-space:

$$\mathrm{TG} : f(\mathcal{M}, \boldsymbol{x}(t_0), \dot{\boldsymbol{x}}(t_0), \ddot{\boldsymbol{x}}(t_0), t) \mapsto \left[\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t), \ddot{\boldsymbol{x}}(t), \boldsymbol{f}(t)\right], \tag{2}$$

hereby the integral relations of $\boldsymbol{x}(t)$, $\dot{\boldsymbol{x}}(t)$ and $\ddot{\boldsymbol{x}}(t)$ must be kept consistent.

In contrast to Finkemeyer and Kröger [7, 10] moving reference frames are integrated here without any restrictions. The whole movement is tied to the

reference frame RF. The pose (hand frame) and the feed forward values may be written as[1]:

$$^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF}} = {}^{\mathrm{RF}^0}\boldsymbol{T}_{\mathrm{RF}} \cdot {}^{\mathrm{BF}}\boldsymbol{T}(\mathcal{M}) \tag{3}$$

$$^{\mathrm{BF}}\begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}(\mathcal{M}) + \boldsymbol{v}_{\mathrm{RF}} + \boldsymbol{\omega}_{\mathrm{RF}} \times \boldsymbol{d}^0_{\mathrm{RF,HF}} \\ \boldsymbol{\omega}(\mathcal{M}) + \boldsymbol{\omega}_{\mathrm{RF}} \end{bmatrix} \tag{4}$$

$$^{\mathrm{BF}}\begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}(\mathcal{M}) + \boldsymbol{a}_{\mathrm{RF}} + \boldsymbol{\alpha}_{\mathrm{RF}} \times \boldsymbol{d}^0_{\mathrm{RF,HF}} + \boldsymbol{\omega}_{\mathrm{RF}} \times \left(\boldsymbol{\omega}_{\mathrm{RF}} \times \boldsymbol{d}^0_{\mathrm{RF,HF}}\right) \\ \boldsymbol{\alpha}(\mathcal{M}) + \boldsymbol{\alpha}_{\mathrm{RF}} \end{bmatrix}, \tag{5}$$

where

$$\begin{aligned} \boldsymbol{d}_{\mathrm{RF,HF}} &= \boldsymbol{R}^T\, {}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{RF}}\boldsymbol{t} - \boldsymbol{R}^T\, {}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF}}\boldsymbol{t} \\ &= \mathrm{Trans}^{-1}\left({}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{RF}}\right) - \mathrm{Trans}^{-1}\left({}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF}}\right). \end{aligned} \tag{6}$$

The movement description in (1) consists of a reference coordinate system $\mathcal{TF}$ and a movement description $\mathcal{MD}$ w.r.t. this reference coordinate system:

$$\mathcal{M} := \left\{ \begin{array}{ll} \mathcal{TF} := \{\mathsf{TFR}, \boldsymbol{x}_{\mathrm{TF}}\} & \text{(Task Frame)} \\ \mathcal{MD} & \text{(Movement Description w.r.t. TF)} \end{array} \right\}. \tag{7}$$

$\mathcal{M}$ may be augmented by a parametrizable command $\mathcal{W}$ which requests other activities than movements to be carried out, e.g. gripper commands or sensor filter adjustments. The *tool command* concept introduced by Finkemeyer [10] is useful for this purpose. It is executed concurrently to the trajectory generation and allows e.g. the gripper to be closed while the robot is moving.

In contrast to robot programming languages established in industrial environments, the execution of skill primitives requires explicit transition conditions to switch from one primitive to another. For this reason the *transition condition* $\mathcal{T}$, a boolean expression, is attached to the primitive. It might contain any available value, e.g. sensor values, drive positions, time constraints. $\mathcal{T}$ is continuously evaluated during the execution of $\mathcal{MP}$. The primitive is considered to be finished when $\mathcal{T}$ becomes true.

2.1 Movement Description and Hybrid Controller Matrix

For the parametrization of the movement a hybrid controller matrix is specified, according to [13]:

$$\mathcal{MD} = \{\mathcal{HC}_{d\times l}, \mathcal{A}\}. \tag{8}$$

$\mathcal{HC}$ denotes the controller matrix, d represents the degrees of freedom, l the number of levels of preemptive controllers. Each element of $\mathcal{HC}$ contains a reference to a control algorithm CN, a setpoint value δ and a vector of additional parameters $\boldsymbol{\Pi}$.

[1] The operators ${}^{(\cdot)}\boldsymbol{T}_{(\cdot)}$ and $\mathrm{RPY}(\cdot)$ generate a homogeneous transformation matrix representing purely translational or rotational transformations (cf. [11]). The nomenclature and additional remarks are contained in [12].

The specification $\mathcal{A} \in \{0; 1\}$ anchors the *task frame* to the hand frame, according to [2].

2.2 Transformations Required for the Task Frame Formalism

According to the *compliance frame* concept the trajectory generation is always calculated w.r.t. the *task frame*, using the hybrid controller matrix. Hence the execution of the primitives introduced above is tied to the *task frame formalism* via $\mathcal{M}$. All sensor data need to be transformed into the *task frame* if possible and meaningfull. After the trajectory calculations are finished, the feed forward branches based on state-space representations require back-transformation into the robots *base frame*. The kinematic transformations necessary for the implementation depend on the controller matrix and the eventual fixation of the *task frame* to the *hand frame*. This may be classified into three cases:

Case 1 (pure positioning): The entries of $\mathcal{HC}$ require position control only. There the task frame is just used to recalculate the new desired pose of the end effector from the reference values $[\boldsymbol{r}_\mathrm{d}\ \boldsymbol{\phi}_\mathrm{d}]^T$. The calculation is done according to

$$^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF,d}} = {}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}} \cdot \mathrm{Trans}(\boldsymbol{r}_\mathrm{d}) \cdot \mathrm{RPY}(\boldsymbol{\phi}_\mathrm{d}). \tag{9}$$

For the planning algorithms operating in the task frame, the following reference values are available:

$$\begin{bmatrix} \boldsymbol{r}_\mathrm{d} \\ \boldsymbol{\phi}_\mathrm{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}^T\, {}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF,d}}\boldsymbol{t} \\ \mathrm{RPY}^{-1}\left({}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF,d}}\right) \end{bmatrix}. \tag{10}$$

Here $\boldsymbol{R}$ and $\boldsymbol{t}$ are used to mask the pure rotation and the pure translation from a homogeneous transformation matrix.

Case 2 (hybrid control, not anchored): The primitive specifies hybrid control, where position, velocity and force control may be active at the same time. The task frame is not attached to the hand frame. The movement starts in the origin of the task frame:

$$^{\mathrm{TF}}\begin{bmatrix} \boldsymbol{r}^0 \\ \boldsymbol{\phi}^0 \end{bmatrix} = \boldsymbol{0}_{6\times 1}. \tag{11}$$

The velocities and accelerations are transformed into the task frame at the beginning of the execution of the primitive:

$$^{\mathrm{TF}}\begin{bmatrix} \boldsymbol{v}^0 \\ \boldsymbol{\omega}^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}^T\left({}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}}\right)^{-1}\boldsymbol{R}\cdot\boldsymbol{v} \\ \boldsymbol{R}^T\left({}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}}\right)^{-1}\boldsymbol{R}\cdot\boldsymbol{\omega} \end{bmatrix} \tag{12}$$

$$^{\mathrm{TF}}\begin{bmatrix} \boldsymbol{a}^0 \\ \boldsymbol{\alpha}^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}^T\left({}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}}\right)^{-1}\boldsymbol{R}\cdot\boldsymbol{a} \\ \boldsymbol{R}^T\left({}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}}\right)^{-1}\boldsymbol{R}\cdot\boldsymbol{\alpha} \end{bmatrix}. \tag{13}$$

Force sensor values are updated continuously by

$$^{\mathrm{TF}}\begin{bmatrix}\boldsymbol{\zeta}\\ \boldsymbol{\tau}\end{bmatrix} = \begin{bmatrix}\boldsymbol{R}^T\,{}^{\mathrm{TF}}\boldsymbol{T}_{\mathrm{SF}}\boldsymbol{R}\cdot\,{}^{\mathrm{SF}}\boldsymbol{\zeta}\\ \boldsymbol{R}^T\,{}^{\mathrm{TF}}\boldsymbol{T}_{\mathrm{SF}}\boldsymbol{R}\cdot\,{}^{\mathrm{SF}}\boldsymbol{\tau}\end{bmatrix}. \tag{14}$$

$^{\mathrm{TF}}\boldsymbol{T}_{\mathrm{SF}}$ is the relative displacement from the sensor frame to the task frame.

During the execution of the primitive, the pose is calculated according to

$$\begin{bmatrix}\boldsymbol{r}\\ \boldsymbol{\phi}\end{bmatrix} = \begin{bmatrix}\boldsymbol{R}^T\,{}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF}}\boldsymbol{t}\\ \mathrm{RPY}^{-1}\left({}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF}}\right)\end{bmatrix}, \tag{15}$$

with

$$\begin{aligned}{}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF}} = {}&{}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}}\cdot\mathrm{T}\left(\left({}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}}\right)^{-1}\,{}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF}}\right)\\ &\cdot\mathrm{Trans}\left({}^{\mathrm{TF}}\boldsymbol{r}\right)\cdot\mathrm{RPY}\left({}^{\mathrm{TF}}\boldsymbol{\phi}\right)\\ &\cdot\mathrm{R}\left(\left({}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}}\right)^{-1}\,{}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF}}\right).\end{aligned} \tag{16}$$

The feedforward branch is transformed by:

$$\begin{bmatrix}\boldsymbol{v}\\ \boldsymbol{\omega}\end{bmatrix} = \begin{bmatrix}\boldsymbol{R}^T\,{}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{TF}}\boldsymbol{R}\cdot\,{}^{\mathrm{TF}}\boldsymbol{v}\\ \boldsymbol{R}^T\,{}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{TF}}\boldsymbol{R}\cdot\,{}^{\mathrm{TF}}\boldsymbol{\omega}\end{bmatrix} \tag{17}$$

The transformation of the acceleration $[\boldsymbol{a}\ \boldsymbol{\alpha}]^T$ and the vector of forces / torques $[\boldsymbol{f}\ \boldsymbol{\tau}]^T$ is carried out in the same way, applying the multiplication of $\boldsymbol{R}^T\,{}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{TF}}\boldsymbol{R}$.

Case 3 (hybrid control in compliance frame concept): The primitive specifies hybrid control and, at the same time, that the *task frame* is anchored to the *hand frame* according to the *compliance frame* concept of Mason [2]. Before a new primitive may be executed, transformations from the previous to the new task frame must be carried out. This is necessary to ensure continuity of the trajectory.

Preparation: Since all hybrid moves are integrating the path w.r.t. the task frame, the pose in the task frame is set to zero. The velocities and accelerations (which are known w.r.t to the task frame of the previous primitive) are moved first so that they act on the origin of the new task frame. Subsequently, their basis is changed so that they are available w.r.t the new task frame. Using the connection vector between the hand frame and the task frame

$$\begin{aligned}\boldsymbol{d}_{\mathrm{TF,HF}} &= \boldsymbol{R}^T\,{}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{TF}}\boldsymbol{t} - \boldsymbol{R}^T\,{}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF}}\boldsymbol{t}\\ &= \mathrm{Trans}^{-1}\left({}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{TF}}\right) - \mathrm{Trans}^{-1}\left({}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF}}\right)\end{aligned} \tag{18}$$

the equations required can be stated:

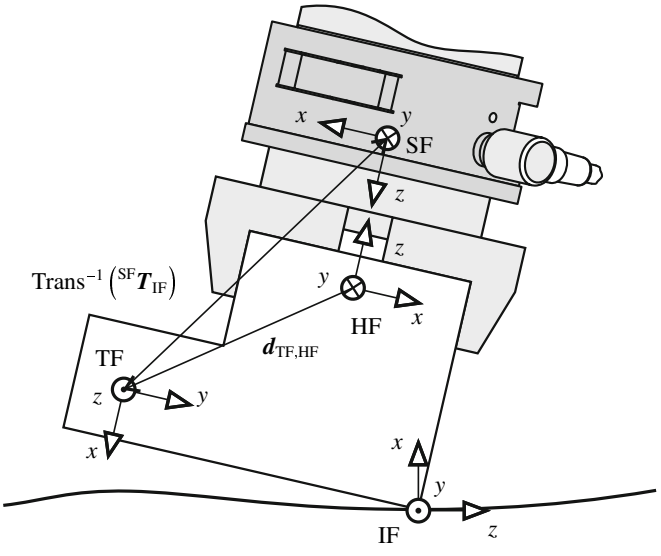


Fig. 1. Coordinate Systems and Vectors involved with Hybrid Moves

$$ {}^{\mathrm{TF}}\boldsymbol{v}^0 = \boldsymbol{R}^T\left({}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}}\right)^{-1}\boldsymbol{R}\cdot\left(\boldsymbol{v}+\boldsymbol{\omega}\times\boldsymbol{d}^0_{\mathrm{TF,HF}}\right) \tag{19} $$

$$ {}^{\mathrm{TF}}\boldsymbol{\omega}^0 = \boldsymbol{R}^T\left({}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}}\right)^{-1}\boldsymbol{R}\cdot\boldsymbol{\omega} \tag{20} $$

$$ \begin{aligned} {}^{\mathrm{TF}}\boldsymbol{a}^0 = {} & \boldsymbol{R}^T\left({}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}}\right)^{-1}\boldsymbol{R}\cdot\left[\boldsymbol{a}+\boldsymbol{\alpha}\times\boldsymbol{d}^0_{\mathrm{TF,HF}}\right. \\ & \left. + \ \boldsymbol{\omega}\times\left(\boldsymbol{\omega}\times\boldsymbol{d}^0_{\mathrm{TF,HF}}\right)\right] \end{aligned} \tag{21} $$

$$ {}^{\mathrm{TF}}\boldsymbol{\alpha}^0 = \boldsymbol{R}^T\left({}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}}\right)^{-1}\boldsymbol{R}\cdot\boldsymbol{\alpha} \tag{22} $$

Calculations during execution: The transformation of the forces / torques measured may be calculated continuously:

$$ {}^{\mathrm{TF}}\boldsymbol{\zeta} = \boldsymbol{R}^T\,{}^{\mathrm{TF}}\boldsymbol{T}_{\mathrm{SF}}\boldsymbol{R}\cdot\,{}^{\mathrm{SF}}\boldsymbol{\zeta} \tag{23} $$

$$ \begin{aligned} {}^{\mathrm{TF}}\boldsymbol{\tau} = {} & \boldsymbol{R}^T\,{}^{\mathrm{TF}}\boldsymbol{T}_{\mathrm{SF}}\boldsymbol{R}\cdot\,{}^{\mathrm{SF}}\boldsymbol{\tau} \\ & + \ \boldsymbol{R}^T\,{}^{\mathrm{TF}}\boldsymbol{T}_{\mathrm{SF}}\boldsymbol{t}\times\left(\boldsymbol{R}^T\,{}^{\mathrm{TF}}\boldsymbol{T}_{\mathrm{SF}}\boldsymbol{R}\cdot\,{}^{\mathrm{SF}}\boldsymbol{\zeta}\right) \end{aligned} \tag{24} $$

The pose is calculated using Eqn. (15), where

$$ {}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF}} = {}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{TF}}\left({}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}}\right)^{-1}\,{}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{HF}}, \tag{25} $$

$$ {}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{TF}} = {}^{\mathrm{BF}}\boldsymbol{T}^0_{\mathrm{TF}}\cdot\mathrm{Trans}\left(\int{}^{\mathrm{TF}}d\boldsymbol{r}\right)\cdot\mathrm{RPY}\left(\int{}^{\mathrm{TF}}d\boldsymbol{\phi}\right). \tag{26} $$

The formulation of most of the equations above is equivalent for both, time continuous domain and time discrete domain. The implementation of these equations is straight-forward. In contrast, a dedicated time-discrete formulation is required for Eqn. (26). We choose the recursive notation:

$$\begin{aligned} {}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{TF}}(kT) = {}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{TF}}(kT - T) \cdot \mathrm{Trans}\left[{}^{\mathrm{TF}}\boldsymbol{r}(kT) - {}^{\mathrm{TF}}\boldsymbol{r}(kT - T)\right] \\ \cdot \mathrm{RPY}\left[{}^{\mathrm{TF}}\boldsymbol{\phi}(kT) - {}^{\mathrm{TF}}\boldsymbol{\phi}(kT - T)\right]. \end{aligned} \tag{27}$$

This creates an incremental behavior as proposed in literature [7]. The velocities acting on the origin of the task frame are expressed w.r.t. the base frame:

$$\left[\begin{array}{c} \boldsymbol{v} \\ \boldsymbol{\omega} \end{array}\right]\Bigg|_{\mathrm{TF}} = \left[\begin{array}{c} \boldsymbol{R}^T\, {}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{TF}}\boldsymbol{R} \cdot {}^{\mathrm{TF}}\boldsymbol{v} \\ \boldsymbol{R}^T\, {}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{TF}}\boldsymbol{R} \cdot {}^{\mathrm{TF}}\boldsymbol{\omega} \end{array}\right]. \tag{28}$$

The calculation of the acceleration $[\boldsymbol{a}\ \boldsymbol{\alpha}]^T\big|_{\mathrm{TF}}$ and the forces / torques $[\boldsymbol{\zeta}\ \boldsymbol{\tau}]^T\big|_{\mathrm{TF}}$ also requires $\boldsymbol{R}^T\, {}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{TF}}\boldsymbol{R}$, and follows the same method as Eqn. (17) and Eqn. (28). Though related to the base frame, the point of attack of these variables is still the origin of the task frame. The connection vector $\boldsymbol{d}_{\mathrm{HF,TF}}$, which is the geometrical inverse of $\boldsymbol{d}_{\mathrm{TF,HF}}$,

$$\boldsymbol{d}_{\mathrm{HF,TF}} = \mathrm{Trans}^{-1}\left({}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{HF}}\right) - \mathrm{Trans}^{-1}\left({}^{\mathrm{BF}}\boldsymbol{T}_{\mathrm{TF}}\right) \tag{29}$$

is used to express the differential variables that attack at the origin of the hand frame relative to the base frame[2]:

$$\boldsymbol{v} = \boldsymbol{v}|_{\mathrm{TF}} + \boldsymbol{\omega}|_{\mathrm{TF}} \times \boldsymbol{d}_{\mathrm{HF,TF}} \tag{30}$$

$$\boldsymbol{\omega} = \boldsymbol{\omega}|_{\mathrm{TF}} \tag{31}$$

$$\begin{aligned} \boldsymbol{a} = {} & \boldsymbol{a}|_{\mathrm{TF}} + \boldsymbol{\alpha}|_{\mathrm{TF}} \times \boldsymbol{d}_{\mathrm{HF,TF}} \\ & + \boldsymbol{\omega}|_{\mathrm{TF}} \times (\boldsymbol{\omega}|_{\mathrm{TF}} \times \boldsymbol{d}_{\mathrm{HF,TF}}) \end{aligned} \tag{32}$$

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}|_{\mathrm{TF}} \tag{33}$$

$$\boldsymbol{\zeta} = \boldsymbol{\zeta}|_{\mathrm{TF}} \tag{34}$$

$$\boldsymbol{\tau} = \boldsymbol{\tau}|_{\mathrm{TF}} + \boldsymbol{\zeta}|_{\mathrm{TF}} \times \boldsymbol{d}_{HF,TF}. \tag{35}$$

For further details regarding the implementation of the *task frame formalism* proposed here the reader is referred to [14].

3 Design Principles for Robot Control Architectures

In order to realize control software for parallel kinematic robots in a safe and efficient way, a specialized design methodology is elaborated. The methodology is

[2] In contrast to Eq. (18), which is only caculated once per primitive, $\boldsymbol{d}_{\mathrm{HF,TF}}$ must be evaluated in every control cycle.

based on the formalism of the previous section. Since the complexity of the integrated models and the corresponding processing time is expected to be considerable, the efficiency of the processing of the data structures during runtime is of particular interest. We identified three paradigms for successfull control software engineering for parallel robots:

(a) Model driven design using *design patterns*
(b) Strict object orientation using active connectors
(c) Function-explicit hierarchy layers

In the following section *design patterns* [15] are presented that result from model based techniques. These patterns are intended to help with the implementation and the integration of new control functionalities. For each particular robot such new functionalities must be developed. Such a development process starts with the elaboration of the design patterns in order to suit a target system or application given.

3.1 Hierarchy and Active Connectors

The analysis of existing control concepts and infrastructures[3] reveals that hierarchical structures increase understandability and flexibility of control architectures. These structures use the definition of the hierarchy layers for one out of several purposes exclusively: It can be found that either organisational relations are represented exclusively, *or* comparable properties of the grouped elements *or* the timing context of the grouped elements. We propose to join these targets and present a new concept which uses explicit connectors to express organisational relations independently from the timing context. First, the control loops can be recognized in the hierarchy we present. Second, as a basic principle, the modules of the layers may be executed concurrently. Third, the architecture is invariant with respect to both the function and the realtime requirements which vary from robot to robot and from application to application.

3.1.1 Context Layers

Based on the insight that priorized execution of control loops can result in superior control performance compared to flat sequential processing it is desirable to integrate appropriate features. In consequence multiple timing contexts should be present in a robot control, which for example provide a level for drive control and levels for higher level control, e.g. position control, human-machine interface. We propose a three-level design, where the execution of a lower level preempts the upper levels. The scheme is displayed in figure 2:

Layer 1 (Lower Layer): This level contains drive control, models, filter and hardware-relevant automata. In its simplest shape it allows the drive control to be

[3] *ORCA2* [16], *Player* [17], *SFMIDDLEWARE* [18], *CLARAty* [19], *MIRO* [20], *OpenRDK*, *MARIE* [21], *MiRPA* [22], *OROCOS* [23], *OSACA* [24], *MCA2* [25]. A discussion of the suitability of these concepts for parallel robots is contained in [14].

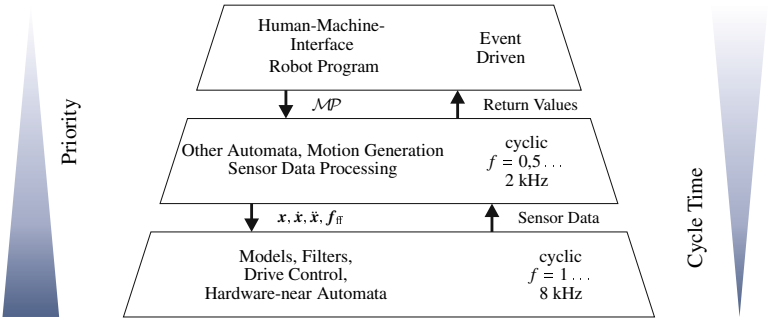


Fig. 2. Hierarchical Task Organisation via Context Layers

started and shut down, as well as eventually necessary referencing schemes to be executed. In extension all functionalities that cannot be commanded suitably by cartesian trajectory data should placed here, e.g. automated configuration changes [26], online calibration or structural vibration control. Due to efficiency maximization the execution of this layer is strictly cyclic.

Layer 2 (Middle Layer): Here sensor data processing and trajectory generation is located. The lower level provides sensor data and asks for trajectory data which include feedforward values.

Layer 3 (Upper Layer): This event-driven layer realizes communication with the user interface, receives movement commands (e.g. skill primitives) and prepares them for further processing.

The data synchronisation between the distinct layers is realized via synchronisation objects that contain double buffers for the objects to be synchronized. This design pattern may be applied generically to cross-layer synchronisation and communication tasks.

3.1.2 Active Connectors

Beside hierarchical context layers that increase abstraction and provide generic solutions for the underlying communication and synchronisation problems a second *design pattern* is provided: So-called *active connectors* help increasing the amount of reusable components. Using connector-based communication increases modularity and visibility of the underlying functionality of the respective modules. We propose the concept of explicit connectors, which means that the layer-membership of a module is uniquely defined by the connector class the base class is derived from. It increases organisational transparency without the need to specify the timing in an early stage.

The systematic of the connector *design pattern* is displayed in fig. 3(a). The connector class encapsulates a behaviour specific to the connector and contains a data basis that allows associated modules to be internally managed with. It may receive commands from outside and return results to there. Internally a connector and its associated modules will share an interface where commands and return values can

be propagated from and to the modules. Additionally the connector class may adress the associated modules via a management interface. Each module may exhibit an interface for external data, which may be directly accessed from outside.

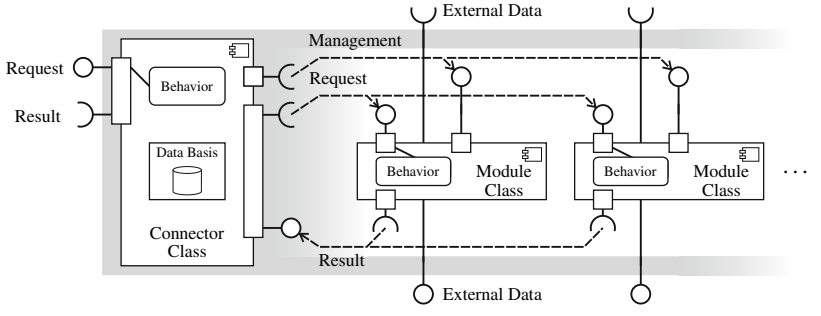
This mechanism supports a developper in that way that he only needs to define the module classes that inherit the interface classes to the connector class, which is shown in the inheritance diagramm 3(b). Then all communication and synchronisation mechanisms are already prepared so that the developper may focus on the algorithm he wants to implement.

3.2 Design Patterns for Modular Sensing and Motion Planning

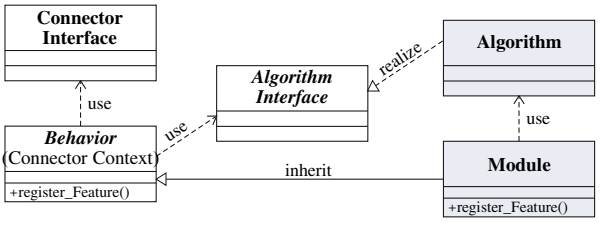
The concept of active connectors is now elaborated in order to establish an architecture for the execution of skill primitives.

3.2.1 Sensor Data Connector

We propose an innovative construct with a paradigm-shifting level of abstraction: The data interface of a sensor only contains three elements. The sensor value itself, a Boolean representing a request for an update of the sensor value and a Boolean that indicates the validity of the sensor value actually present. A sensor module offers this data interface directly to the motion generation modules. The activation of a particular sensor module is executed by the connector module. Each time the



(a) Systematic of Active Connectors



(b) Class Diagramm of a Module in Connector Context

Fig. 3. Active Connector *Design Pattern* and Realization of Module

connector is activated it checks which sensor modules shall be activated in order to process a request or update its sensor value.

3.2.2 Motion Planning Connector

For motion planning algorithms a connector similar to the sensor connector may be used to refine the general active connector design pattern. On activation due to a new primitive the controller matrix $\mathcal{MD}$ is handed over to the connector. After a validity check the control information requested is generated. The control information contains a list of motion algorithms that require activation in cyclic schedule.

On activation each motion module calculates the trajectory information required and returns this data via the result interface to the connector. There the data of the motion modules involved is joined together and the calculations of the *task frame formalism* are carried out. The result of this calculation is published in the trajectory interface of the connector.

In case an error occurs in this scheme the connector executes an algorithm which computes a trajectory for an emergency stop. It ensures that the brake trajectory stops the robot with respect to cartesian coordinates instead of drive coordinates.

3.2.3 Synthesis

The design patterns for sensor data processing and motion generation introduced above may be joined formally to fully describe the execution of skill primitives (figure 4(a)): The connectors for sensor modules and motion modules are part of the trajectory generation. Hence they form a modular architecture for motion generation.

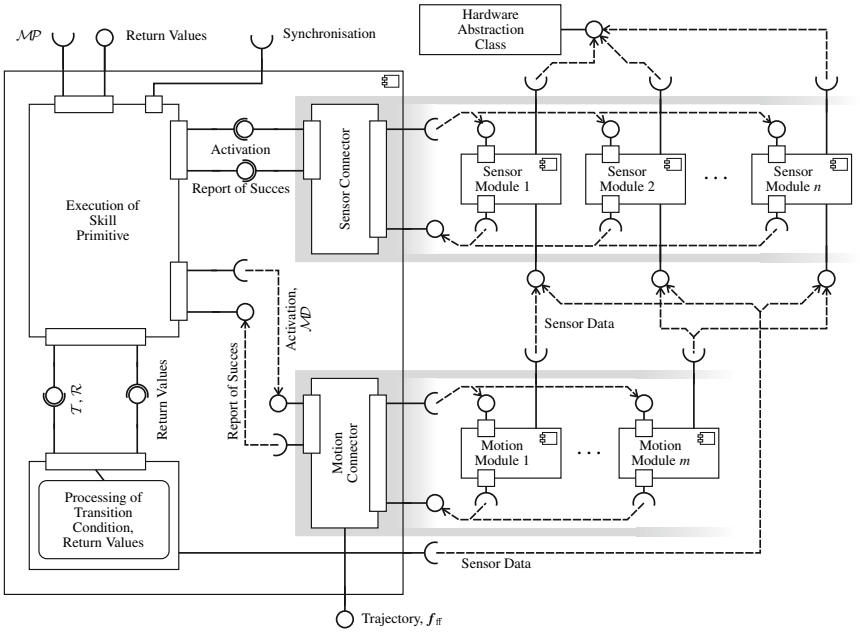
There is an execution unit which operates the connectors and another component which processes the transition condition. The sensor modules use their external interfaces to access the sensor hardware and forward these data to the motion modules and the component that processes the transition condition. The motion modules are employed to generate the trajectory. The cyclic overall activity to execute the skill primitive is displayed in figure 4(b).

4 RCA562 - An Extendable Robot Control Architecture for Parallel Kinematic Robots

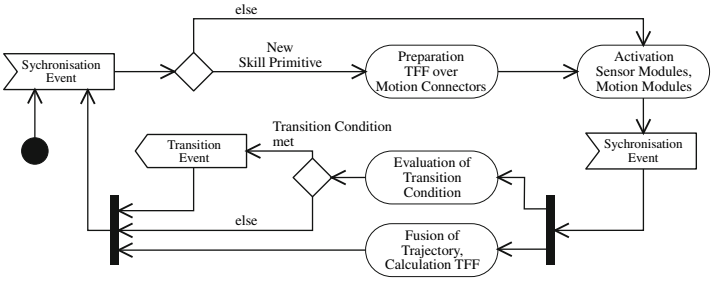
The formal specification outlined in the sections above was implemented in the context of the *Collaborative Research Center 562*. The application was realized to control the HEXAII and the TRIGLIDE robot[4] and is called *RCA562*[5]. There the possibilities provided by the formal specification regarding self management, security and scalability have been used extensively. For additional information about the software engineering techniques used in *RCA562* and the underlying middleware

[4] The robots HEXAII andTRIGLIDE were mechanically designed by Budde [27].

[5] **R**obot **C**ontrol **A**rchtitecture of Collaborative Research Center **562**.



(a) Connector View on Trajectory Generation (simplified)



(b) Cyclic Activity of Skill Primitive Execution

Fig. 4. Trajectory Generation: Connector Systematic and Cyclic Activity

MIRPA-X we refer to [28]. Insights of the drive control scheme implemented in *RCA562* are presented in [29].

4.1 Special Features

RCA562 features some special functionality that are described in the following.

Process Modularisation: Spreading of functionalities across multiple processes yields huge advantages compared to single-process applications. This was realized based on the middleware MiRPA-X [30]. Since the core functionalities

may be encapsulated against the non-core functionalities errors may be handled actively by the core in order to ensure safety.

Dynamic Communication Pathes: When a communication middleware is used, dynamical communication pathes become available. Then, always the control algorithm which is most suitable at a given instant may be used. Additionally the scheduling is adjusted in each cycle instant which minimizes computation load and hence increases reactivity of the processes located in the upper part of the hierarchy. This is carried out implicitly by the activation of the connector which only activates the modules required.

Scalability and Distribution: Theoretically processes that do not share data dependencies may be executed concurrently, e.g. on two parallel cores in a multi-core environment. *RCA562* with its built-in concurrency model supporting dynamic scheduling inherently supports such functionality. Even distribution across multiple computers is derivable from the models of activity and behaviour. Methods used for analysis of realtime software and the examination of the properties are presented in [31, 32]. The effort related with the implementation of such schemes is motivated by the computational load that parallel robots require. For example, complex sensor modules, such as image processing, or complex motion modules that navigate in huge amounts of data may be swapped out.

Reusability: *RCA562* was designed to feature as much reusable components as possible. The result can be seen from the fact that currently two different robots (the HEXAII and the TRIGLIDE robot) are maintained out of the same repository. The high level of code reusability was confirmed by LaQuSo, which evaluated *RCA562* [33] and resumed that "no significant code duplicates were found".

Safety: *RCA562* features an outstanding high level of safety. The connector models that keep the connectors reactive in every situation play a key role in this aspect. In case a module connected does not respond to a request transmitted by a connector the connector may withdraw the activity from this module and execute a redundant algorithm that ensures secure fall-back operation. Additionally the modularization and hierarchical structure simplifies detection and tracing of errors.

Based on its modularity, RCA562 benefits from contributions throughout the SFB562 and features some more functionalities especially dedicated to parallel robots. Among these are *automated change of working mode* [26], autonomuous calibration of the kinematic model [34], avoidance of singularities [35], and adaptronics [36].

4.2 Validation Scenarios

The advantages of the calculation of the complete dynamic state in the *task frame formalism* for parallel robots were validated in experiments. The calculation of feedforward values in the *task frame* and their correct transformation to the robot's base frame pay off in smooth feedforward branches. This increases the performance of model based drive control, a detailed presentation is given in [14]. In the work cited,

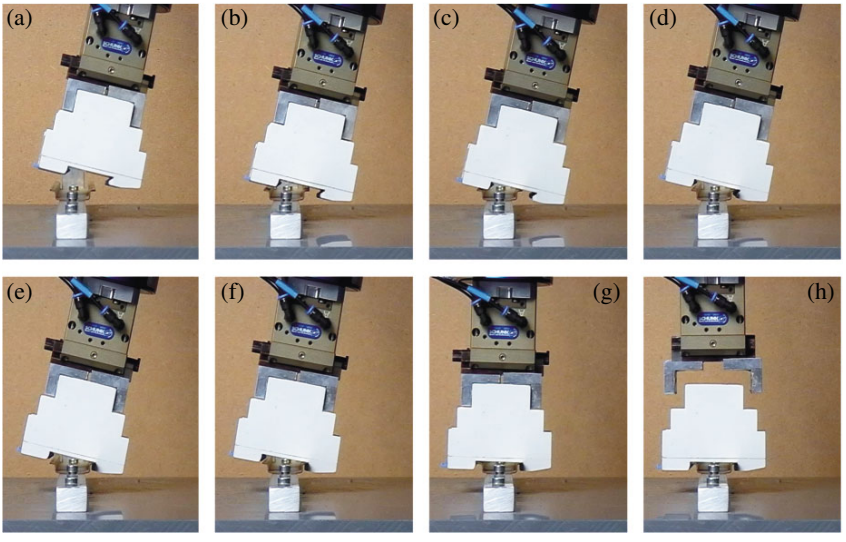


Fig. 5. Assembly Sequence involving Hybrid Control and Skill Primitives

the effectiveness of the described design principles and the capabilities of *RCA562* are also covered.

Here, an example is presented which gives an impression of the effectiveness of skill primitives and hybrid control in the *task frame formalism*: Electric jacks that are used in switching cabinets shall be mounted on a top hat rail. This task is extraordinarily demanding for automated assembly systems because of its wide tolerances and its snap-action mechanism that must be assembled with a rotary movement about an a-priori not exactly known rotating axis. Figure 5 shows the assembly sequence where a contact is established using hybrid control at first (line-plane contact, (a)-(c)). Then the edge is seeked by moving perpendicular to the contact plane until the corner is reached (d). There the *interaction frame* is estimated, and a rotation towards the final pose is started ((e)-(g)). Errors in the frame estimation are compensated by suitable force control. When a predefined torque threshold is exceeded, the assembly is finished.

There were many people of the SFB562 who contributed to this example, using the kit character of *RCA562*. For example, the automated generation of skill primitive nets based on contact information graphs is discussed in [37]. In this way this exemplary application was realized in a time efficient way which would by far not be achieveable with conventional methods.

5 Concluding Remarks

Software architectures for powerful control for robotic systems in general, but especially for parallel kinematic robots, face conflicting requirements: The user demands

a generic interface for task description which masks away the specialties of a particular robot structure. The tasks that are assigned to robots require hybrid control in moved reference frames. The control developer asks for component reusability across multiple robot structures on one hand. On the other hand he needs flexibility which lets him implement robot specific functionalities efficiently. For best maintainance a hierarchical organisation where both, functional and time contexts, can easily be recognized is desirable.

Each of these requirements is covered in at least one existing architecture, but unfortunately none of these architectures satisfies all of them. This gave us the motivation to work out *design patterns* that are powerful enough to consider all of the aspects mentioned above. Based on this toolkit *RCA562* has been designed, a control architecture customized to the high level of model integration for parallel robots. *RCA562* offers a generic programming interface using the *task frame formalism*, is open for new components to be plugged in during runtime, provides clarity and features as much reusable components as possible (which was confirmed by external code evaluation). In the end it also provides elevated safety by decoupling and dynamic scheduling of those components that are absolutely indispensable and those that are optional. But *RCA562* also represents a powerful infrastructure which is widely neutral with respect to specific applications and which benefits from parallel exexution in a multi-threaded environment.

In resume the design methods and implementation methods may be employed generically to face control tasks in robotics. In this context our application to parallel robot structures may be considered as a validation scenario of the generic set of *design patterns* which were presented.

Acknowledgements. The authors wish to thank the German Research Foundation (DFG) for the support of this work. Also, the authors are grateful for the work of Nicky Gerritsen und Jeroen Veldhuis (LaQuSo), who evaluated the quality of the code of *RCA562*.

References

1. Gosselin, C.M.: On the Design of Efficient Parallel Mechanisms. In: Angeles, J. (ed.) Computational methods in mechanical systems. NATO ASI seriesSeries F, Computer and systems sciences, vol. 161, pp. 68–96. Springer, Heidelberg (1998)
2. Mason, M.T.: Compliance and force control for computer controlled manipulators. IEEE Trans. on Systems, Man and Cybernetics, 418–432 (1981)
3. de Schutter, J., van Brussel, H.: Compliant robot motion I: A formalism for specifying compliant motion task. Int. Journal of Robotics Research 7(4), 3–17 (1988)
4. Hasegawa, T., Suehiro, T., Takase, K.: A Model-Based Manipulation System with Skill-Based Execution. IEEE Transactions on Robotics and Automation 8(5), 535–544 (1992)
5. Finkemeyer, B., Borchard, M., Wahl, F.: A robot control architecture based on an object server. In: IASTED International Conference Robotics and Manufacturing (RM 2001), Cancun Mexiko, pp. 36–40 (2001)
6. Thomas, U., Finkemeyer, B., Kröger, T., Wahl, F.M.: Error-tolerant execution of complex robot tasks based on skill primitives. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 3069–3075 (2003)

7. Kröger, T., Finkemeyer, B., Thomas, U., Wahl, F.M.: Compliant motion programming: The task frame formalism revisited. In: IEEE International Conference on Mechatronics and Robotics, pp. 1029–1034 (2004)
8. Kröger, T., Finkemeyer, B., Wahl, F.M.: A task frame formalism for practical implementations. In: IEEE International Conference on Robotics and Automation, pp. 5218–5223 (2004)
9. Merlet, J.-P.: Parallel Robots. Kluwer Academic Publishers, Dordrecht (2000)
10. Finkemeyer, B.: Robotersteuerungsarchitektur auf der Basis von Aktionsprimitiven. Shaker Verlag, Aachen (2004)
11. Sciavicco, L., Siciliano, B.: Modelling and Control of Robot Manipulators. McGraw-Hill, London (2000)
12. Maaß, J., Reisinger, T., Hesselbach, J., Schumacher, W.: A versatile robot control architecture for sensor integrated assembly. In: Proceedings of ISR/Robotik 2006 (2006)
13. Reisinger, T.: Kontaktregelung von Parallelrobotern auf Basis von Aktionsprimitiven. Ph.D. thesis, TU Braunschweig (2008) (in german)
14. Maaß, J.H.: Ein Beitrag zur Steuerungstechnik für parallelkinematische Roboter in der Montage. Ph.D. thesis, Fakultät für Maschinenbau, TU Braunschweig, Vulkan Verlag (2009) (in German)
15. Selic, B.: Architectural patterns for real-time systems. In: UML for real: design of embedded real-time systems, pp. 171–188. Kluwer Academic Publishers, Norwell (2003)
16. Brooks, A., Kaupp, T., Makarenko, A.: ORCA: Components for robotics Internet-Ressource (2008), `http://orca-robotics.sourceforge.net/`
17. The player project Internet-Ressource (2008), `http://playerstage.sourceforge.net/` Free Software tools for robot and sensor applications
18. Jorg, S., Nickl, M., Hirzinger, G.: Flexible signal-oriented hardware abstraction for rapid prototyping of robotic systems. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3755–3760 (2006)
19. NASA and JPL. Claraty robotic software (2008), `http://claraty.jpl.nasa.gov`
20. Utz, H., Sablatnog, S., Enderle, S., Kraetzschmar, G.: Miro-Middleware for Mobile Robot Applications. IEEE Transactions on Robotics and Automation 18(4), 493–497 (2002)
21. Côté, C.: Mobile and autonomous robotics integration environment (2008), `http://marie.sourceforge.net`
22. Finkemeyer, B., Kröger, T., Kubus, D., Olschewski, M., Wahl, F.M.: Mirpa: Middleware for robotic and process control applications. In: Workshop on Measures and Procedures for the Evaluation of Robot Architectures and Middleware. IEEE International Conference on Intelligent Robots and Systems, San Diego, USA, pp. 76–90 (2007)
23. Bruyninckx, H.E.A.: OROCOS - open source robot control software (2008), `http://www.orocos.org/` IST-2000-31064
24. The OSACA Project Consortium. Open system architecture for controls within automation systems (2010), `http://www.osaca.org/`
25. FZI Karlsruhe. Modular controller architecture version 2 Internet-Ressource (2008), `http://www.mca2.org/` Abteilung Interaktive Diagnose- und Servicesysteme
26. Budde, C., Helm, M., Last, P., Raatz, A., Hesselbach, J.: Configuration Switching for Workspace Enlargement. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 175–189. Springer, Heidelberg (2010)
27. Schütz, D., Budde, C., Raatz, A., Hesselbach, J.: Parallel Kinematic Structures of the SFB 562. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 109–124. Springer, Heidelberg (2010)

28. Maaß, J., Hesselbach, J., Kohn, N.: Open Modular Robot Control Architecture for Assembly Using the Task Frame Formalism. International Journal of Advanced Robotic Systems 3, 001–010 (2006)
29. Kolbus, M., Reisinger, T., Maaß, J.: Robot Control based on Skill Primitives. In: Robotics and Applications: Sixth IASTED International Conference Proceedings (2005)
30. Dadji, Y., Michalik, H., Kohn, N., Steiner, J., Beckmann, G., Möglich, T., Varchmin, J.U.: A Communication Architecture for Distributed Real-Time Robot Control. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 213–231. Springer, Heidelberg (2010)
31. Maaß, J., Steiner, J., Amado, A., Hesselbach, J., Huhn, M., Raatz, A.: Self-management in a control architecture for parallel kinematic robots. In: Proc. ASME IDETC/CIE, Brooklyn, New York, USA (2008)
32. Steiner, J., Goltz, U., Maaß, J.: Self-Management within a Software Architecture for Parallel Kinematic Machines. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 355–371. Springer, Heidelberg (2010)
33. Gerritsen, N., Veldhuis, J.: Maintainability assessment of robot software TU Braunschweig. Technical report, TU Eindhoven (2007)
34. Last, P., Raatz, A., Hesselbach, J.: Calibration of Parallel Kinematic Structures - Overview, Classification and Comparison. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 93–106. Springer, Heidelberg (2010)
35. Dietrich, F., Maaß, J., Bier, C., Pietsch, I., Raatz, A., Hesselbach, J.: Detection and Avoidance of Singularities in Parallel Kinematic Machines. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 77–92. Springer, Heidelberg (2010)
36. Algermissen, S., Sinapius, M.: Robust Gain Scheduling for Smart-Structures in Parallel Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 159–174. Springer, Heidelberg (2010)
37. Thomas, U.: Automatisierte Programmierung von Robotern für Montageaufgaben. Ph.D. thesis, Fakultät für Mathematik und Informatik, TU Braunschweig (2008) (in german)

Assembly Planning and Task Planning — Two Prerequisites for Automated Robot Programming

Ulrike Thomas and Friedrich M. Wahl

Abstract. This contribution aims to present a complete process chain starting from the initial specification of assembly tasks via assembly sequence planning, all the way through task planning and finally task execution. It demonstrates how robot programs can be generated automatically from CAD data. For assembly sequence planning, a new assembly planning system has been developed and evaluated with several representative assembly groups. As an interface to robot control systems, manipulation primitives have been introduced. Manipulation primitive nets are an appropriate way to deal with uncertainties occurring during assembly task execution. This contribution introduces a new approach, demonstrating how to generate manipulation primitive nets automatically. Firstly, CAD data are segmented into surface primitives. A contact formation graph based on topological contacts between such surface primitives is generated afterwards. Based on these contact formation graphs, manipulation primitive nets can be successfully derived. The concept for automated robot programming presented in this contribution is supported by real experiments of actual assembly tasks.

1 Introduction

In this paper, two essential requirements for automated robot programming are considered: assembly planning and task planning. In industry, products are often required to be adapted with significant high rates in a very short period of time.

Ulrike Thomas
German Aerospace Center (DLR), Institut of Robotics and Mechatronics,
P.O.Box 1116, 82234 Weßling, Germany
e-mail: ulrike.thomas@dlr.de

Friedrich M. Wahl
Technische Universität Braunschweig, Institute for Robotics and Process Control,
Mühlenpfordtstraße 23, 38106 Braunschweig, Germany
e-mail: f.wahl@tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 333–354.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

Markets demand more and more sophisticated products produced with increasing speed. To obtain this high rate of productivity, it requires the application of flexible machine tools like robots. If high production rates need to be achieved, parallel robots are often an optimal choice due to their excellent dynamical properties and their high level of precision. Nevertheless, to overcome uncertainties during the execution of robot programs, sensors should be integrated intensely, which increases the programming complexity of such systems. Since labor is one of the main contributors to product costs, the reduction of human interaction in robot programming is both a general and significant aim in manufacturing. One possibility to overcome these dramatically increasing product costs is to deploy tools and software for automated robot programming. On the one hand, CAD systems were developed and came on the market already during the last decades; hence, highly accurate product models are available. Despite this, the model world consisting of robot models, product models, relative poses of objects etc. still suffers from errors often caused by model tolerances as well as by pose tolerances. On the other hand, robot control systems became much more sophisticated, e.g., they often employ a variety of different sensors like force/torque sensors, 2d or even 3d vision sensors, or distance sensors. For the application of force/torque sensing, the main underlying sensor integration concept is based on Mason's compliance frame approach [1], where each DOF can be controlled independently according to different sensor feedbacks. Raibert and Craig were some of the first, who implemented such a flexible control scheme [2]. De Schutter et al. developed a closed formalism — the task-frame formalism, which can be applied very intuitively [3]; moreover, he introduced a control concept, which implements the task frame formalism [4]. Kathib suggested to use selection matrices to switch between different controllers [5]. Hasegawa developed skill primitives and applied an appropriate control scheme for easy and certain robot tasks like placing an object onto a table [6]. Since then, many research groups have been developing interfaces for robots and their control systems to enable robots to interact in a more and more sophisticated manner with their environments. In the beginning of the Collaborative Research Center 562, Mosemann and Wahl suggested to decompose automatically planned robot tasks into pre-defined sequences of skill primitives [7]. In [8], we gave a more formal specification of skill primitive nets and demonstrated the execution with a control scheme developed by Finkemeyer [9]. This skill primitive concept was further developed by contributors of the Collaborative Research Center 526 [10, 11, 12]. To execute skill primitive nets a new control scheme for parallel robots has also been described in [12]. In this paper, skill primitives are denoted as manipulation primitives[1]. Over the last decade, control units became ready to be used with automatically generated robot programs at least in research laboratories. For the automation of robot programming, two prerequisites — assembly planning and task planning — are required in order to map computed plans to the appropriate control units.

[1] This name is chosen due to the fact that manipulation primitives are simpler manipulations of objects than skills are. The skill of a robot is obtained by a net of manipulation primitives.

First developments in assembly planning started with the systems LAMA and AUTOPASS [13, 14]. Both systems map abstract assembly commands to robot programs, which are then executed by commercial robot programming systems. In the late eighties, further assembly planning systems have been implemented. In Edinburgh, the RAPT-system was created by Popplestone and Ambler [15]. They applied symbolic spatial relations to specify relative object poses. Within their software, a geometric as well as a kinematic engine was implemented in order to obtain feasible assembly paths. Also Levi developed a system, called APROM, which applies symbolic spatial relations from which the system computes assembly sequences. Levi's assembly planner requires additional knowledge about geometric feasible assembly sequences, provided in form of precedence graphs by the user [16]. KAMRO is an extension of the APROM-system. KAMRO is able to generate programs for assembly with two or more robots working in parallel. This is achieved by mapping precedence graphs to Petri-nets. Another early assembly planning system is LEGA, which was developed by Bourjaul [17]. It uses interactive questions answered by construction engineers in order to gather geometrical constraints. The most sophisticated systems have been the Archimedes 2 and the HighLAP systems. The Archimedes system applies so-called directional blocking graphs and non-directional blocking graphs respectively to extract geometric constraints. Assembly task sequences are mapped to V+ programs [18, 19, 20]. HighLAP computes contact surfaces between objects and derives geometrical assembly constraints from the known contacts [7]. It evaluates assembly sequences with many various criteria. Knoll et al. developed a system able to understand commands like "insert the red screw into the blue cube" and it maps such commands to robot control systems [21]. The evaluation of the system was done with simple toy objects. Schwarzer and Schweikard introduced in their assembly planner a new algorithm, which solves the geometric reasoning problem by formulating it as an optimization problem. They obtained good results by the application to rather simple objects [22]. If objects become more complex, i.e., they are modeled with many thousands triangles, the solution space will increase rapidly. All these systems have in common that almost no considerations of uncertainties were taken into account. Ostrovsky and Joskowicz developed a system, which deals with uncertainties, but it has only been applied to assemblies in 2d [23]. However, demands still exist for more accurate and robust computations of assembly sequences.

Another challenge is the generation of robot programs for sensor based execution. Almost all solutions are based on classified geometrical contacts, which have been introduced by Laugier [24]. For the representation of possible contact states, contact formation graphs have been suggested by Hirai [25]. Xiao and Ji [26, 27] introduced an algorithm for the generation of contact formation graphs. Nodes in contact formation graphs represent possible contact states between two objects during assembly execution. The edges represent possible transitions from one contact state to another. For example, when an object comes in touch with a table, a vertex/face contact formation likely exists. It can be transformed by a rotation into an edge/face contact, and so on. It is difficult to recognize such contact states during assembly,

because sensor noise appears as always. Lefebvre applied nonlinear Bayesian filtering techniques for reliable contact identification during assembly execution [28]. Meeussen tried to integrate the planning process into the execution process and suggested the usage of a particle filter to identify contact states. His concept has been evaluated only with simple objects [29]. As can be seen from our above considerations, further research in automated robot programming is still necessary.

In the next section, we describe our specification interface based on CAD-models, which provides a comfortable way to specify what to assemble. Section three contains a detailed explanation about our new assembly sequence planner as well as results for real industrial products. In section four, we describe how the concept of contact formation graphs can be extended in order to consider higher order surfaces, too. We describe a new algorithm to transform contact formation graphs into manipulation primitive nets, which can be loaded into robot control systems. The last section of our paper gives a conclusion.

2 Specification of Assembly Tasks

For the specification of assembly tasks, CAD models of objects, robots, fixture units as well as the kinematic description of robots are required. In addition, a precise description of available sensors is also needed; in our case, we employ a 6d force/torque sensor. Furthermore, we need specifications of object poses in the final state, in particular relative transformations of the objects to be assembled. For the specification of relative object poses, symbolic spatial relations were suggested by Ambler and Popplestone [15]. We developed a graphical user interface, which facilitates the easy definition of such symbolic spatial relations. For this aim, challenges are the computation of unique relative poses, the detection of contradictions in the specification as well as the recognition of the left DOFs from a set of symbolic spatial relations. In order to specify a relation, the user has to select features by marking appropriate object surfaces and to choose a type of the symbolic spatial relation. Fig. 1 shows some examples of specified symbolic spatial relations for the assembly group of a valve. In this example, the user has chosen the relation *fits* and the appropriate features *shaft* and *hole* (first line). For each of such relation, a linear matrix equation can be deduced. Thus, the underlying algebra system obtains a set of equations given in general by

$$ {}^{feature_j}\mathbf{T}_{object_j} \cdot {}^{object_j}\mathbf{T}_{W} \cdot {}^{W}\mathbf{T}_{object_i} \cdot {}^{object_i}\mathbf{T}_{feature_i} \cdot \mathbf{T}_{relation_k} = \mathbf{I} \tag{1} $$

where $T_{relation_k}$ is substituted according to the k-th relation with

$$ T_{relation} := \begin{cases} Trans(0, y_k, z_k) \cdot Rot_x(\alpha_k) \cdot Rot_y(\pi) & against \\ Trans(0, y_k, z_k) \cdot Rot_x(\alpha_k) & coplanar \\ Trans(x_k, y_k, z_k) \cdot Rot_x(\alpha_k) & parallel \\ Trans(x_k, 0, 0) \cdot Rot_x(\alpha_k) & aligned \\ Trans(x_k, 0, 0) \cdot Rot_x(\alpha_k) \cdot Rot_y(\pi) & fits \\ Trans(x_k, 0, 0) \cdot Rot_x(\alpha_k) & collinear \end{cases} \tag{2} $$

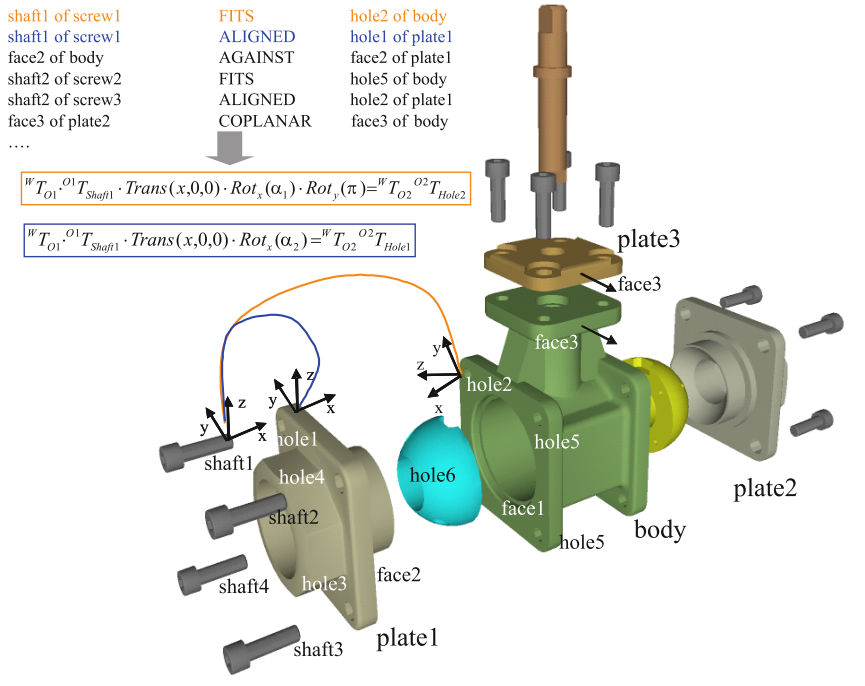


Fig. 1. A part of the specification of a valve assembly with some symbolic spatial relations.

All equations are transformed, such that they are equal to zero. This system of equations must be solved regarding to the unknowns, which are caused by the symbolic spatial relations and the object poses represented by the set

$$\mathcal{X} := \{y_k, z_k, x_k, \alpha_k, \ldots, pose_x, pose_y, pose_z, pose_{\phi_x}, pose_{\phi_y}, pose_{\phi_z} \ldots\} \quad (3)$$

An algorithm for finding solutions for all unknowns is applied as follows:

(a) Deduce a matrix equation for each symbolic spatial relation, as mentioned above.

(b) Substitute each irrational expression with a rational expression to achive highest accuracy by keeping the correlation between *sinus* and *cosinus* persistent, because Gröbner-Bases are only definied over rational expressions. It is implemented by searching for two values u and v for which

$$\sin\alpha \approx \frac{u^2 - v^2}{u^2 + v^2} \qquad \cos\alpha \approx \frac{2uv}{u^2 + v^2} \quad (4)$$

yields.

(c) Substitute each *sinus* and *cosinus* expression with place holders a and b also add $a^2 + b^2 = 1$ to the system of equations.

(d) Compute a Gröbner Basis by Buchberger's algorithm. If Buchberger's algorithm reduces at least one equation so that it equals to 1 a contradiction is detected, otherwise solve the system according to the unknowns.

Based on this underling algebra system, the following requirements are met: Those relations that lead to contradictions in the specification are detected. Also, remaining DOFs are shown to the user. Object poses are computed according to the given specification. With such a graphical-interactive tool, complete assembly tasks can be specified in a comfortable way. The next section, describes how assembly sequences are computed upon a given specification.

3 Assembly Sequence Planning

It is a long-term aim to generate and evaluate assembly sequences automatically with a sophisticated assembly planning system. The accurate computation of geometric constraints strongly influences a reliable assembly plan. While generating assembly plans, a difficulty to deal with is the existing exponential number of sequences. For example, if an assembly contains n parts and it should be separated into two disjunct groups, already $2^{n-1} - 1$ possibilities exist. For assembly sequence planning, $\frac{3^n+1}{2} - 2^n$ assembly sequences have to be analyzed in the worst case. Even the 2d assembly planning problem is NP-hard [30]. Because an assembly planning system can benefit highly from an efficient geometric reasoning test, we spent much effort on the development of such a geometric algorithm described in more detail in the next subsection. In our assembly planner, we generally apply the well-known assembly-by-disassembly strategy. We start with the specification of the complete assembly group obtained as described above. Step-by-step assembly parts or subassemblies are removed and analyzed if a mating operation is geometrically possible or not. By reversing this order of disassembly operations, we obtain a geometrical feasible assembly plan. Further constraints effecting the assembly plan are considered. Therewith, we obtain optimal assembly plans with respect to certain evaluation criteria.

3.1 Solving the Geometric Reasoning Problem Efficiently

Our computation of possible mating directions is based on configuration space obstacles, which makes it very accurate and robust. Lozano-Perez introduced configuration space obstacles [31], but since then no precise solution in 3d was available for real assembly parts modeled by thousands of triangles. This was due to its difficulties in decomposing 3d objects into convex parts. It has been shown, that the decomposition of a part into a minimal number of convex subparts is NP-hard. But for generation of 3d configuration space obstacles, it is sufficient to decompose object surfaces only. Charzelle introduced an algorithm [32] for the decomposition of surfaces into convex sub-patches. Thus, we implemented this algorithm and accelerated its speed by the usage of 3d hierarchies. Fig. 2 middle illustrates a result of our implementation for a headlight lamp with its projection unit.

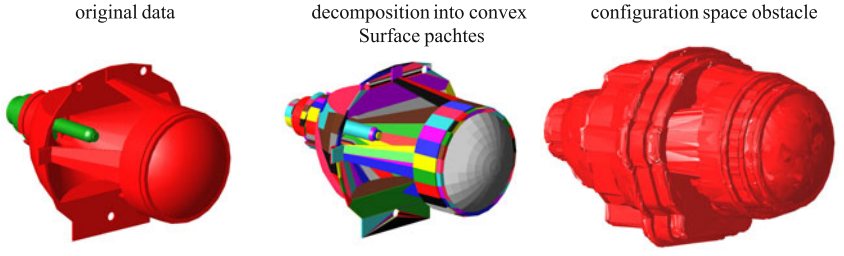


Fig. 2. Computation of configuration space obstacles. On the left side original data — the lamp (green) and the projection unit (red) — are shown; in the middle surfaces decomposed into convex patches are depicted, and on the right side, the generated configuration space obstacle is illustrated.

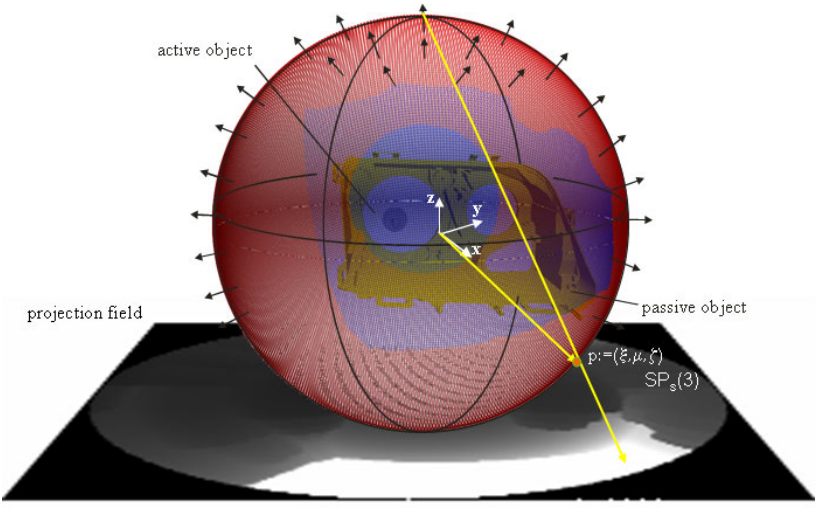


Fig. 3. Stereographical projection of configuration space obstacles.

Based on the set of sub-patches for object A, $\mathcal{S}_A := \{s_1^A, \ldots, s_n^A\}$, and object B, $\mathcal{S}_B := \{s_1^B, \ldots, s_m^B\}$, the Minkowski-Point-Set is computed by pairwise subtraction of vertex coordinates belonging to each convex patch: $\mathcal{M}(s_i^A, s_j^B) := \{\boldsymbol{v}_k \ominus \boldsymbol{v}_l | \ \forall \boldsymbol{v}_k \in s_i^A, \ \forall \boldsymbol{v}_l \in s_j^B\}$. Building the convex hull over the pairwise computed Minkowski-Point-Set results into the configuration space obstacle for one pair of patches CO_{s^A,s^B}. Repeating this for all pairs of patches (s_i^A, s_j^B) leads to the complete configuration space obstacle $CO_{A,B}$. This computationally intensive procedure has the advantage that it only needs to be computed once for each pair of assembly objects. The right of Fig. 2 illustrates the result for the computation of the configuration space obstacle for the headlight lamp with the projection unit. The occupied space

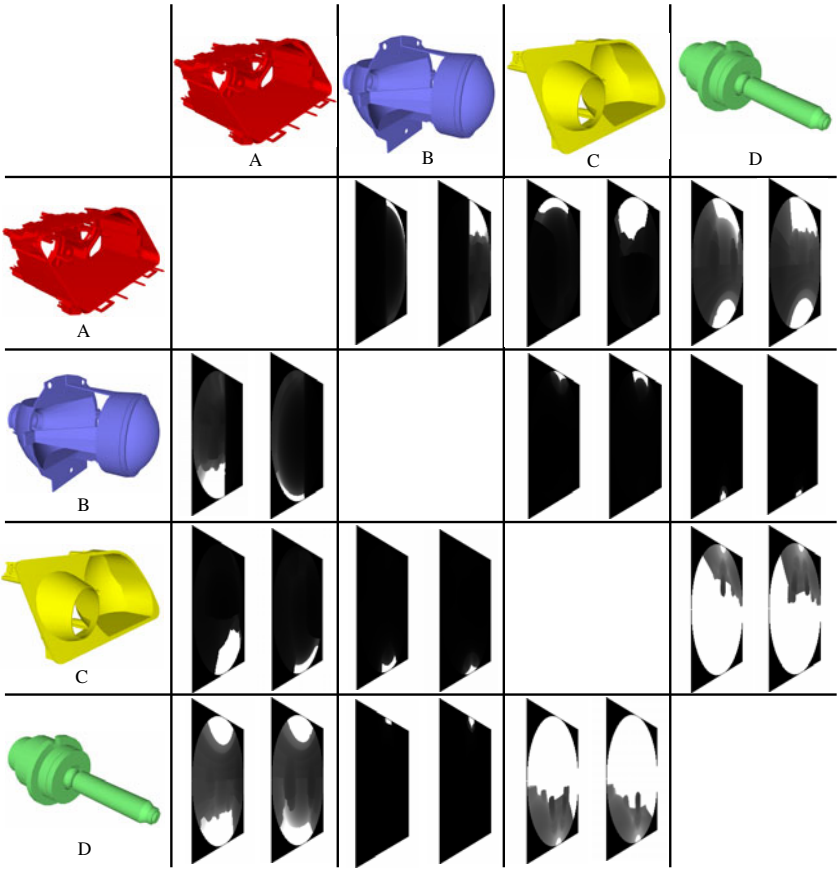


Fig. 4. The look-up table computed in the preprocessing step.

represents a collision of objects. The border of the configuration space object, as it is well known, is equivalent to all configurations, where the two objects are in contact. In order to extract feasible mating directions, the configuration space obstacle is projected onto a surrounding sphere. Each uncovered point on the sphere's surface represents a geometrically feasible mating direction. Hence, these projections are very important for analyzing assembly cuts during the computationally intensive disassembly process. For saving all mating directions efficiently, a $2\frac{1}{2}$d image is generated by a stereographic projection, (Fig. 3). The images generated by this mapping are stored in a look-up table for each pair of objects belonging to the assembly, as it is illustrated for some parts of a headlight example in Fig. 4.

The value at image coordinates $w|_{x,y}$ represents the distance d between two objects along direction $\boldsymbol{m} := (\xi, \eta, \zeta)^T$. Regarding to the stereographical projection, $\boldsymbol{m}$ is given by

$$\boldsymbol{m} := \left(\frac{2x}{x^2+y^2+1}, \frac{2y}{x^2+y^2+1}, \frac{x^2+y^2-1}{x^2+y^2+1} \right)^T \tag{5}$$

In order to avoid singularities of the stereographical mapping, two images are used, one for each half of the sphere. If the lower half sphere is referred, $\boldsymbol{m}'$ is obtained by multiplication with $(1, 1, -1)^T$. Thus, translational geometrical feasible mating directions can be deduced efficiently. Altogether, for each pair of objects, two binary images $BSP_{N/S}$ are defined as follows

$$BSP_{N/S}|_{x,y}(2) = \begin{cases} 1 \; if \;\; \boldsymbol{m} \;\; or \; respectively \;\; \boldsymbol{m}' \;\; is \; free \\ 0 \; else \end{cases} \tag{6}$$

N denotes the upper half of the sphere, and S the lower half of the sphere respectively. As mentioned above, the projection fields need only be computed once for each pair of objects. Furthermore, they are collected in the look-up table. Fig. 4 illustrates such a look-up table for a small assembly consisting of four parts. For the investigation of a possible disassembly step in $\{A, D\}$ and $\{B, C\}$, the look-up table can be applied as follows: The images of the lookup-table belonging to the pairs (B, A), (B, D), (C, A), and (C, D) are concatenated only by an AND-conjunction. If the resulting images includes at least one entry, which is equal to 1, a mating direction for this (dis)assembly step can be found (cf. Fig. 5). Formally, the geometric reasoning test can be described as follows: Assuming an assembly $\mathcal{A} = \{p_1, \ldots, p_m, \ldots, p_n\}$ has to be separated into the sub-groups $\mathcal{A}_1 = \{p_1, \ldots, p_m\}$ and $\mathcal{A}_2 = \{p_{m+1}, \ldots, p_n\}$, the AND-conjunction of images given by $BSP_{N/S}(2)(p_i, p_j)$ with $p_i \in \mathcal{A}_1$ and $p_j \in \mathcal{A}_2$ needs to be considered. Thus, the function Sep, which evaluates the geometric reasoning problem for two sub-assemblies $\mathcal{A}_1$ and $\mathcal{A}_2$ is given by

$$Sep(\mathcal{A}_1, \mathcal{A}_2) := \begin{cases} true \;\; \exists \, x, y \in [-1, 1] \mid \bigwedge_{i=1}^{m} \bigwedge_{j=m+1}^{n} BSP_{N/S}|_{x,y}(2)(p_i, p_j) = 1 \\ false \; else \end{cases} \tag{7}$$

In this way, a very efficient geometric reasoning test can be implemented, because the computation is only based on binary images. Thereby, a fast and robust test is established with highest precision obtained from the computation of configuration space obstacles. During the generation and evaluation of assembly sequences, the assembly planner takes a great benefit from such an efficient test. In addition, uncertainties in assembly models are admitted. They are treated by an additional tolerance value. This value is assigned to each convex hull and, hence, influences the size of each configuration space obstacle resulting in a tolerance belt. Thus, tolerances of objects are considered for the computation of mating directions. Despite objects are modeled face to face, mating directions can be found [33]. This is an important issue for applying the algorithm to real industrial assembly groups, as it can be seen in the next section.

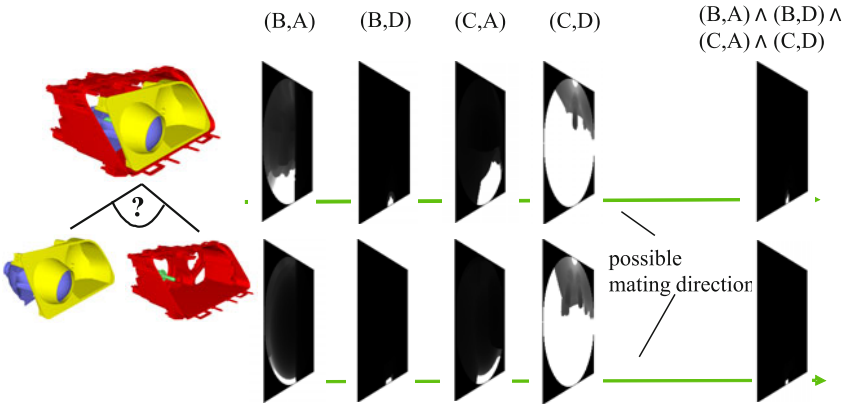


Fig. 5. An example for the geometric test.

3.2 Generation and Evaluation of Assembly Sequences

For the generation and evaluation of assembly sequences it is obvious, that not all sequences can be considered due to their exponential number. Thus, good sequences should be found with high probability as early as possible. For this reason, a connectivity graph is introduced. In this graph, the nodes represent objects; they are considered to be connected if the distance between two objects is less than a predefined threshold. This is a very weak definition of connectivity, but it ensures that good sequences are not excluded too early. Fig. 6 illustrates the two applied methods. On the left side, a 2d assembly is illustrated by means of a shell-based model. On the right side, the connectivity graph is shown according to the illustrated assembly group. For the generation of assembly sequences, two heuristics are applied. While employing the shell-based heuristic, those assembly cuts are found whose removed objects belong to the outer shells. In the example on the left side of Fig. 6, the removal of object 6 is investigated prior to the removal of object 5, 1 and so on. Furthermore, assemblies exist, for which it is still impossible to find feasible solutions by applying a shell-based model. Therefore, a second strategy based on the spectral-bi-partitioning of graphs is applied. In the example depicted on the right side, the cuts of the assembly group into $(1,3,5)/(2,4,6)$, $(1,3,4)/(2,5,6)$, and $(2,3,4/1,5,6)$ is computed by such a spectral bi-partitioning of graphs. The utilization of both strategies ensures that a sufficient number of various assembly cuts is found. With the bi-partitioning of graphs equally balanced sub-assemblies are generated. Based on such a balanced partition, further cuts are investigated by changing the object's memberships, i.e., a part is removed from one sub-assembly and assigned to the other one according to its connectivity. The algorithm investigates and evaluates various cuts of one sub-assembly by applying this strategy recursively.

For the evaluation of assembly sequences hard and soft constraints need to be analyzed. Some hard constraints are the geometrical feasibility and the connectivity of sub-assemblies, such that no parts drop during assembly. These two criteria are

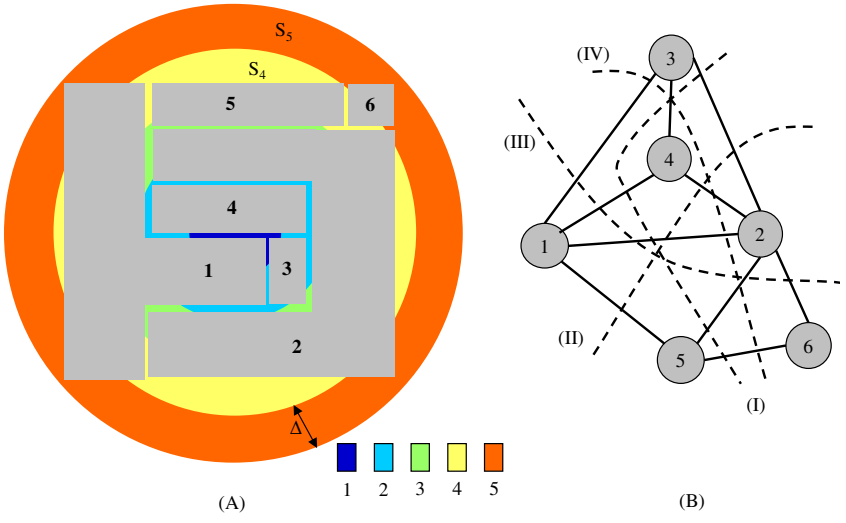


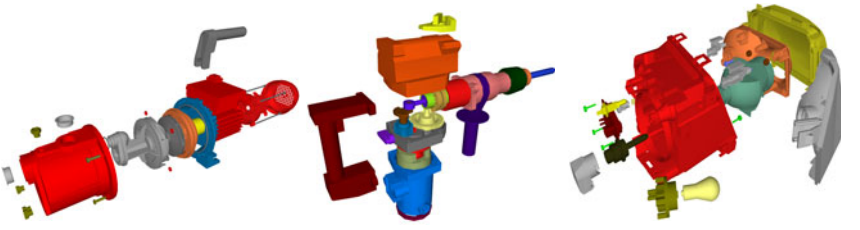
Fig. 6. Left: A shell-based model. Right: The connectivity graph of the group shown at the left side and three cuts from the calculated extended spectral bi-partitioning.

addressed by the algorithms described above. One additional criterion for assembly operations is the possibilities for grasping. If no suitable grasp for a subassembly can be found, the operation is discarded. All successfully analyzed assembly cuts are recursively inserted into an AND/OR-graph [34]. By traversing the AND/OR-graph from bottom to top, valid assembly sequences are evaluated by weighted sums. The following criteria are taken into account for the evaluation. Each function obtains one or two assembly steps in the assembly sequence as input and returns a numerical value as output.

- Geometrical accessibility: That means the size of the geometric accessible area on top of the projection sphere is considered.
- Parallelism: The cut in the AND/OR-graph obtains less costs if the sub-assemblies are about the same size, e.g., the number of parts in the subgroups should be balanced.
- Mechanical stability: Each sub-assembly should be mechanical stable. We applied a heuristic, which is based on differential potential energy fields computed for some gravitation vectors.
- Manageability: For evaluation of manageability the number of geometrically possible grasps for subgroups are considered. Valid grasping configurations are inserted manually up to now.
- Reorientation: Reorientations should be minimized during assembly if reorientation can not be avoided high costs are taken into account.
- Change of grasping tools: It is at least one grasp with an appropriate tool defined for each object. If it is necessary to change the tool very often, the costs for this assembly step increases.

- Change of mating directions: If the direction of mating must be changed very often between two assembly steps, the costs are very high.

An optimal assembly plan is found by a branch-and-bound search. Regarding the above mentioned evaluation criteria, the assembly sequence is selected on the basis of minimum costs estimated by a weighted sum. Fig. 7 illustrates some benchmarks for our assembly planner and Fig. 8 shows a computed assembly plan for the headlight assembly.



(A) (B) (C)

Fig. 7. Benchmarks for our assembly planning system: (A) water pump, 45 parts, 94 min. preprocessing, 45 min. planning. (B) Drilling machine, 23 parts, 29 min. preprocessing and 9 min. planning and (C) automotive headlight, 33 parts, 67 min. preprocessing and 15 min. planning. The computation for preprocessing steps are executed on an SGI Octane 2 and the results for planning are obtained on a Pentium 4 with 2.8 GHz processor.

4 Task Planning

For the execution of manipulation primitives, appropriate control systems have been developed. For reading about this topic, please refer to [35, 36]. Manipulation primitives serve as interface between the planning described here and the control system, which is described in more detail in [37]. Here, a short overview about manipulation primitive nets and their application is given.

For placing an object onto a table, Fig. 9 illustrates an excerpt of such a manipulation primitive net. The nodes contain the manipulation primitives, which basically consist of a task frame specification, elements as set-point values for the controllers, and a stop condition. A manipulation primitive is executed by an appropriate controller as long as the stop condition can not be evaluated to be true. When the stop condition is true, the next manipulation primitive is loaded into the controller, according to the outgoing edges of the last node in the manipulation primitive net. One big challenge to fill the gap between planning and execution is the development of an algorithm to generate such nets for advanced assembly tasks automatically.

Fig. 8. A computed assembly plan for an automotive headlight assembly.

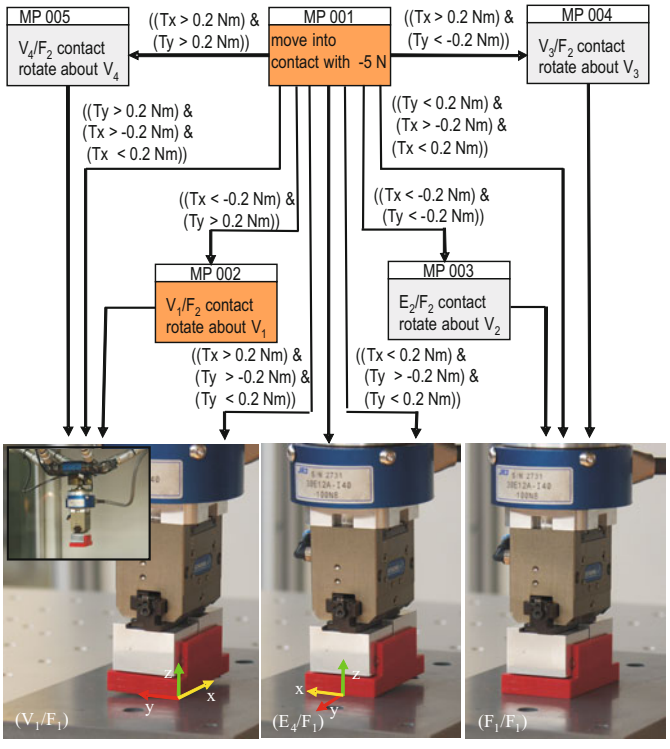


Fig. 9. An excerpt of a manipulation primitive net for the assembly task *placing an object on top of a table*.

4.1 Contact Formation Graphs

Contact formation graphs constitute a basis for the generation of manipulation primitive nets. Well-known contact states between two rigid objects are classified into *vertex/face, edge/face, edge/edge, face/face* according to [26]. In our work, we extend the contact states by contacts between higher-order surfaces. Therefore, we need the definition of topological elements: A topological element *te* is one of the elements in TE with

$$te \in TE := \{vertex(V), edge(E), circular\ arc(A), face(F), \\ cylindrical\ surface(C), spherical\ surface(S)\} \tag{8}$$

If objects are modeled by triangle meshes, a topological element is a segment of triangles. A contact between two topological elements te_i and te_j is defined according to [26]. It is distinguished between a main contact and an elemental contact. A main contact $(te_i, te_j) \in TE_A \times TE_B$ is a pair of topological elements, where te_i belongs to object A and t_j belongs to object B. A main contact can be expressed by a set of elemental contacts, but an elemental contact cannot be expressed by other contacts. In

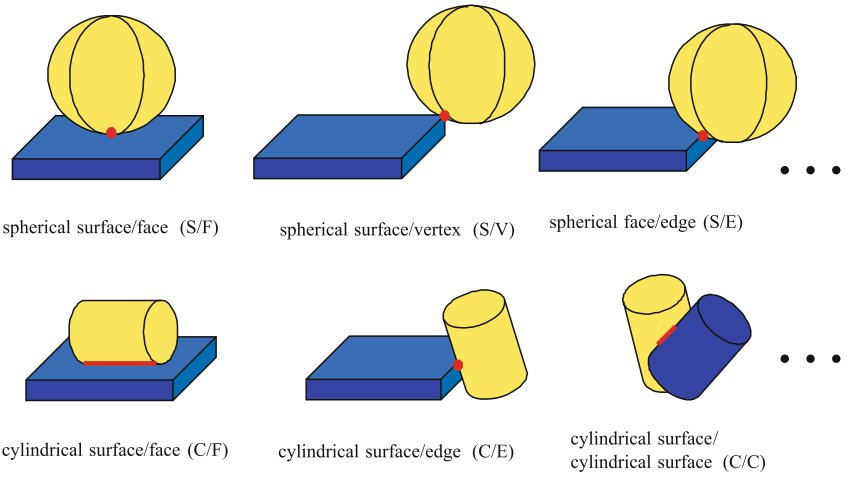


Fig. 10. Some examples of the contact formations for higher order surfaces.

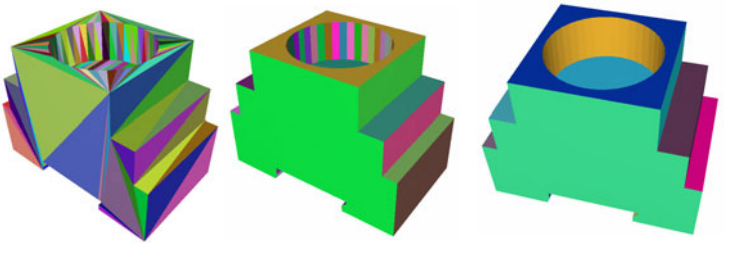


Fig. 11. Left: Original data (triangle mesh); middle: segmentation into planes, and right: segmentation into cylindrical and spherical surfaces.

addition to the well-known contact formations from [24, 26, 38], further contact formations regarding to higher order surfaces (spherical and cylindrical surfaces) have been applied here. Fig. 10 illustrates some examples of the used contact formations.

Altogether, the new introduced contact formations are in particular: cylindrical face/ vertex, cylindrical arc/face, cylindrical arc/edge, cylindrical face/edge, cylindrical arc/cylindrical arc, cylindrical arc/cylindrical face, cylindrical face/face, cylindrical face/cylindrical face, spherical face/face, spherical face/vertex, spherical face/edge, spherical face/cylindrical face, spherical face/cylindrical arc, spherical face/spherical face. For the generation of contact formation graphs, the topological elements have to be recognized from CAD data first, actually from 3d triangle meshes. Therefore, a segmentation algorithm has been implemented, which generates, for example, results as illustrated in Fig. 11. Each object is segmented, and all topological elements are extracted according to our definition given above.

A contact formation graph CFG consists of two sets CF and T. The set CF contains all nodes and contact formations, and the set T represents the transitions from one contact state to another. Thus, a transition $t_{ij} := (CF_i, CF_j)$ between two contact formations is included in the graph if CF_i and CF_j are neighbors. Two contact formations are neighbors if either $CF_i \subset CF_j$ or $CF_j \subset CF_i$ is satisfied. The relation $CF_i \subset CF_j$ is given if all reducible main contacts in CF_i are included in CF_j. If this condition is true for two contact formations CF_i and CF_j, an edge can be inserted into the graph. During the generation of contact formation, multiple contacts occur. Thus, prior to analyzing the neighborhood, all generated contact formations have to be reduced according to given rules, such that only elemental contacts remain. Here, due to lack of space, just some of the rules are listed:

$$\mathcal{CF} = \begin{cases} (V_1/F_2) \wedge (F_1/F_2) \Rightarrow (F_1/F_2) & if\ V_1 \in F_1 \\ (E_1/F_2) \wedge (F_1/F_2) \Rightarrow (F_1/F_2) & if\ E_1 \in F_1 \\ (A_1/F_2) \wedge (F_1/F_2) \Rightarrow (F_1/F_2) & if\ A_1 \in F_1 \\ (V_1/F_1) \wedge (V_2/F_1) \Rightarrow (E_1/F_1) & if\ V_1 \in E_1 \wedge V_2 \in E_1 \\ (A_1/C_2) \wedge (C_1/C_2) \Rightarrow (C_1/C_2) & if\ A_1 \in C_1 \\ (A_1/C_2) \wedge (A_2/C_1) \Rightarrow (C_1/C_2) & if\ A_1 \in C_1 \wedge A_2 \in C_2 \\ \ldots \end{cases} \tag{9}$$

For example: If a vertex/face contact and a face/face contact exist, they can be reduced to a face/face contact if the vertex belongs to the face, which is already in contact. The same applies if the cylindrical arc of object A belongs to the cylindrical face of object A and a cylindrical arc/cylindrical face as well as a cylindrical face/cylindrical face contact occur. If the condition for the neighborhood of two contact formations is fulfilled, an edge is inserted into the contact formation graph. By applying these rules, the depicted contact formation graph of Fig. 12 is generated for the robot task *placing a power socket on top of a hat rail* shown in Fig. 14.

4.2 From Contact Formation Graphs to Nets of Manipulation Primitives

For generation of manipulation primitive nets, rules have to be applied in order to transform nodes and edges of a contact formation graph into respective manipulation primitives and transitions. In addition to these rules, a look-up table is used for the selection of appropriate set-point values. Each edge in the contact formation graph represents a change of the contact state, which could be carried out by one manipulation primitive. But consider the following example: Given the contact state called *free* and the succeeding contact states in Fig. 12; only one manipulation primitive lets the contact situation switch from *free* into all succeeding contact states. Thus, just one manipulation primitive is necessary. Each contact state is annotated by a set of configurations. Therewith, the algorithm searches for a transformation from free space into all other succeeding contact states, which can be reached according to the inertial object pose. In the given example, a single translational transformation is found. The free space node has five succeeding contact states, thus five transitions are needed to decide, in which situation the assembly task is after the execution of

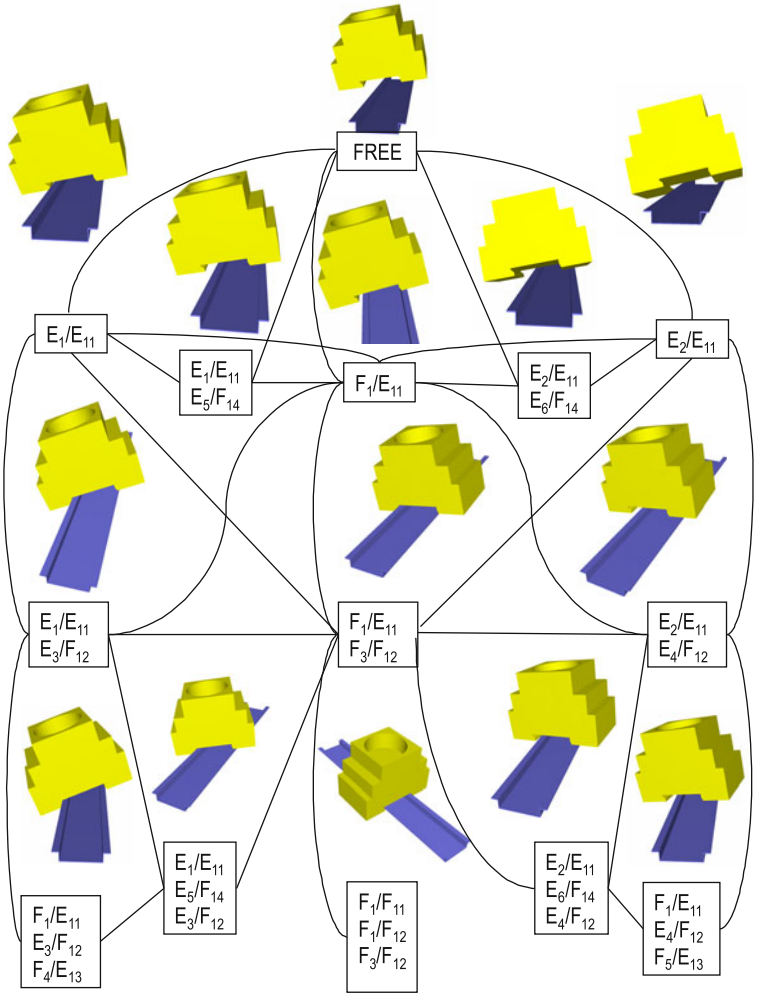


Fig. 12. A generated contact formation graph for the power socket to be assembled on top of the hat rail.

the first manipulation primitive. These distinguishable equations are obtained by assuming a translational movement and by investigation of possible contact regions. For each contact region, a unique equation is derived. Transitions in the manipulation primitive net are annotated with these equations. For choosing appropriate values for the task frame and the controller, probable contact points are extracted, and transformations switching from one contact state to the others are configured with appropriate set-point values. Therewith, the manipulation primitive net is

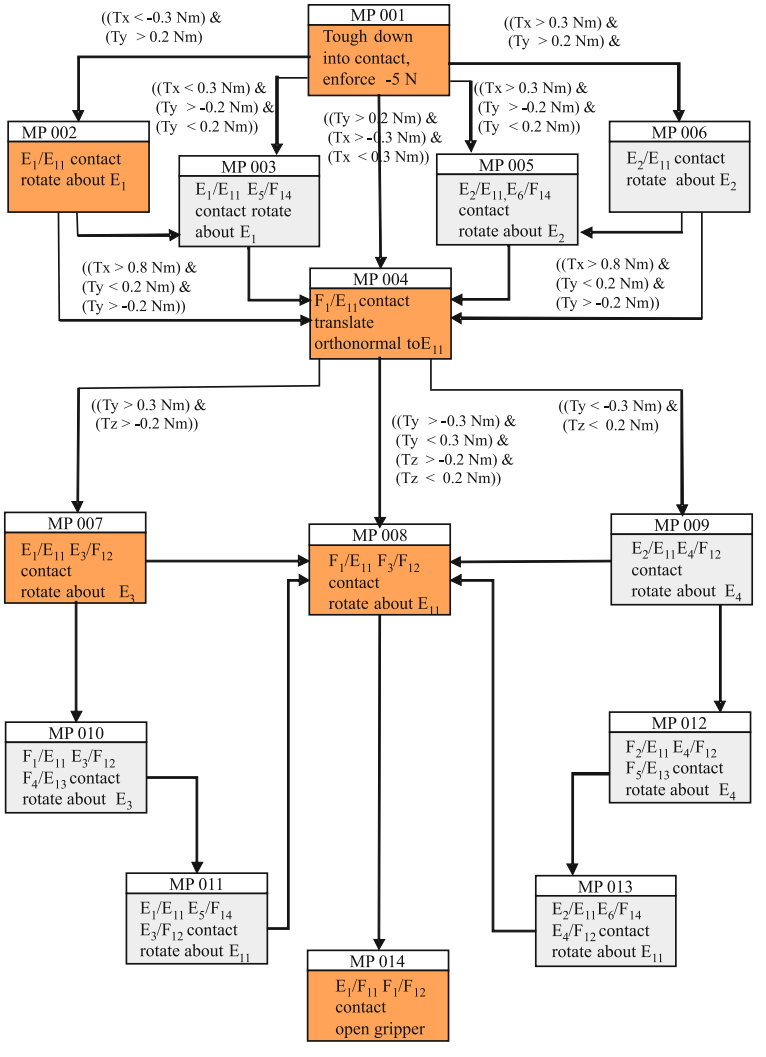

MP 001
Tough down into contact, enforce -5 N
((Tx < -0.3 Nm) & (Ty > 0.2 Nm)
((Tx < 0.3 Nm) & (Ty > -0.2 Nm) & (Ty < 0.2 Nm))
((Ty > 0.2 Nm) & (Tx > -0.3 Nm) & (Tx < 0.3 Nm))
((Tx > 0.3 Nm) & (Ty > -0.2 Nm) & (Ty < 0.2 Nm))
((Tx > 0.3 Nm) & (Ty > 0.2 Nm) &
MP 002
E1/E11 contact rotate about E1
MP 003
E1/E11 E5/F14 contact rotate about E1
MP 005
E2/E11 E6/F14 contact rotate about E2
MP 006
E2/E11 contact rotate about E2
((Tx > 0.8 Nm) & (Ty < 0.2 Nm) & (Ty > -0.2 Nm))
MP 004
F1/E11 contact translate orthonormal to E11
((Tx > 0.8 Nm) & (Ty < 0.2 Nm) & (Ty > -0.2 Nm))
((Ty > 0.3 Nm) & (Tz > -0.2 Nm))
((Ty > -0.3 Nm) & (Ty < 0.3 Nm) & (Tz > -0.2 Nm) & (Tz < 0.2 Nm))
((Ty < -0.3 Nm) & (Tz < 0.2 Nm)
MP 007
E1/E11 E3/F12 contact rotate about E3
MP 008
F1/E11 F3/F12 contact rotate about E11
MP 009
E2/E11 E4/F12 contact rotate about E4
MP 010
F1/E11 E3/F12 F4/E13 contact rotate about E3
MP 012
F2/E11 E4/F12 F5/E13 contact rotate about E4
MP 011
E1/E11 E5/F14 E3/F12 contact rotate about E11
MP 013
E2/E11 E6/F14 E4/F12 contact rotate about E11
MP 014
E1/F11 F1/F12 contact open gripper

Fig. 13. The generated manipulation primitive net for the robot task: socket on top of a hat rail.

generated step by step on the basis of the contact formation graph. Fig. 13 illustrates the so generated manipulation primitive net. In the next section, experimental results are shown when such a net is loaded into the control unit of our parallel manipulator HEXA.

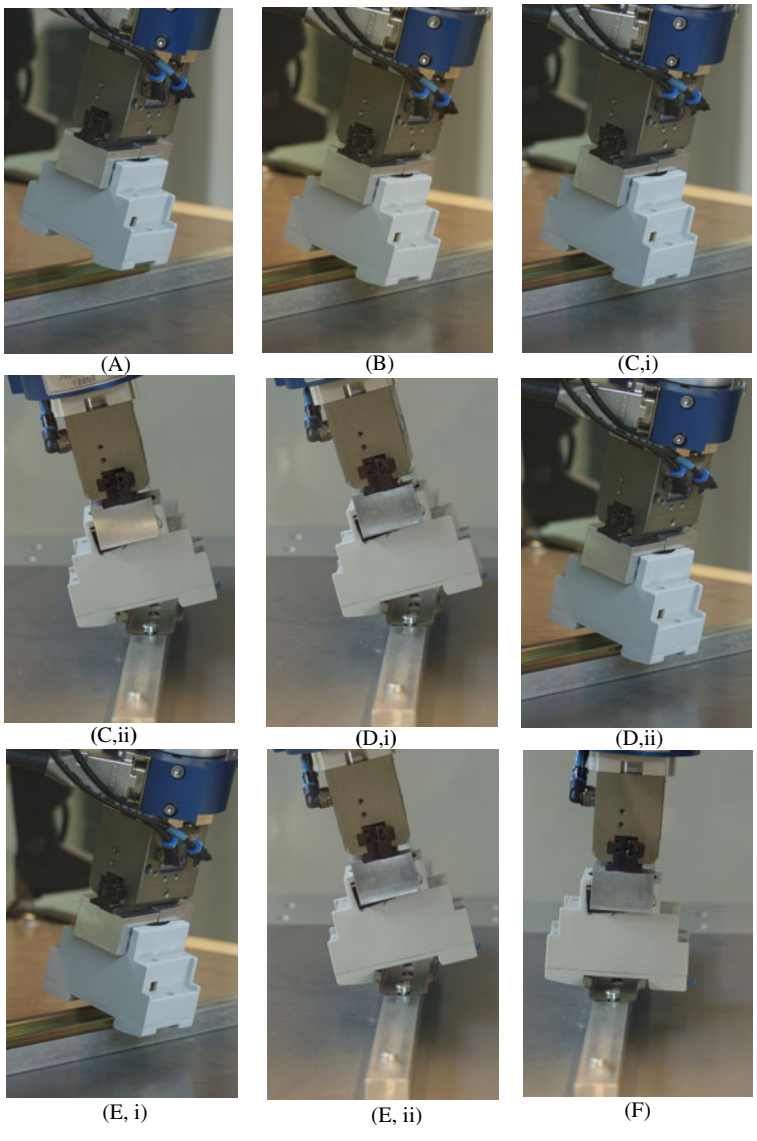


Fig. 14. Steps during the execution of the manipulation primitive net of the robot task *power socket on top of a hat rail*.

4.3 Experiments

The assembly task *put a power socket on top hat rail* is depicted in Fig. 14. The execution is carried out by the control system described in [37]. The situations, in which the colored manipulation primitives in Fig. 13 are executed, are illustrated in Fig. 14. First of all, with a translational movement, the socket comes in contact with

the hat rail, and in our example an edge/edge contact is established (MP 001). With the evaluation of the equation $(T_x < -0.3\,Nm) \wedge (Ty > 0.2\,Nm)$ the contact E_1/E_{11} is recognized (cf. image B in Fig. 14). Now, with the next manipulation primitive (MP 002), the robot rotates the socket about edge E_1 until a surface/edge contact F_1/E_{11} is established (cf. Ci and Cii in Fig. 14).

With a translational orthogonal movement to edge E_{11}, the next contact state $E_1/E_{11}, E_3/F_{12}$ is ensured and recognized by evaluating the equation $((Ty > 0.3\,Nm) \wedge (Tz > -0.2\,Nm))$ (cf. Fig. 14 (D i) and (D ii)). Subsequently, a rotation about the edge E_3 is executed, such that a further rotation about E_{11} is necessary in order to assemble the socket on top of the hat rail successfully (cf. image (E i) and (E ii)). The longest edge of the hat rail is now collinear with the edge E_{11}. Finally, the power socket is rotated about this edge as long as the measured torque is less as a determined threshold of $2,5\,Nm$. When the stop condition is evaluated to be true, the socket is fixed on top of the hat rail (cf. Fig. 14, image (F)).

5 Conclusion

In this contribution, a complete process chain from the specification of assembly tasks to the execution of robot tasks has been described. For a more detailed description, please refer to [33]. With the graphical-interactive specification of robot tasks expensive labor costs can be reduced. An assembly planning system has been developed, which is able to generate geometrical feasible assembly sequences. Also, the system can evaluate these sequences according to important criteria. The assembly planner is able to deal with real industrial assemblies and computes executable assembly sequences, as it has been shown here. To overcome the gap between planning and execution, we introduced an algorithm for the generation of manipulation primitive nets. This algorithm first generates contact formation graphs, which are used to derive manipulation primitives, automatically. The manipulation primitive nets are executed by a parallel robot (HEXA). Further experiments will be conducted to demonstrate the power of the concept in the near future. We have also restricted our algorithms to rigid bodies; experiments as well as concepts for deformable objects will be established in the future. Another interesting topic is the integration of more sensors and the automated planning of their usage. For example, some questions are worth to investigate: Where do cameras have to be placed, or which vision algorithms are suitable for which robot task? These sensor tasks could also be planed by a sophisticated system, and it demands further research in this very interesting field.

Acknowledgements. We are grateful to the German Research Foundation for supporting the Collaborative Research Center 562. Also, we would like to thank QNX-Software-Systems for providing licenses of their real time operating system.

References

1. Mason, M.T.: Compliance and force control for computer controlled manipulators. IEEE Trans. on Systems, Man, and Cybernetics 11, 418–432 (1981)
2. Raibert, M.H., Craig, J.J.: Hybrid position/force control of manipulators. ASME Journal of Dynamic Systems, Measurement and Control 102, 126–133 (1981)
3. Schutter, J., van Brussel, J.: Compliant robot motion II. A control approach based on external control loops. The International Journal of Robotics Research 7(4), 18–33 (1988)
4. Schutter, J., van Brussel, J.: Compliant robot motion I. A formalism for specifying compliant motion tasks. The International Journal of Robotics Research 7(5), 3–17 (1988)
5. Khatib, O.: A unified approach for motion and force control of robot manipulators: The operational space formulation. IEEE Journal of Robotics and Automation, RA 3(1), 43–53 (1987)
6. Hasegawa, T., Suehiro, T., Takase, K.: A model-based manipulation system with skill-based execution. IEEE Trans. on Robotics and Automation 8(4), 535–544 (1992)
7. Mosemann, H., Wahl, F.M.: Automatic decomposition of planned assembly sequences into skill primitives. IEEE Trans. on Robotics and Automation 17(5), 709–718 (2001)
8. Thomas, U., Finkemeyer, B., Kröger, T., Wahl, F.M.: Error-tolerant execution of complex robot tasks based on skill primitives. In: Proc. of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, vol. 3, pp. 3069–3075 (2003)
9. Finkemeyer, B.: Robotersteuerungsarchitektur auf der Basis von Aktionsprimitiven (in German). Shaker Verlag, Aachen (2004)
10. Kröger, T., Finkemeyer, B., Thomas, U., Wahl, F.M.: Compliant motion programming: The task frame formalism revisited. In: Mechatronics and Robotics, Aachen, Germany, pp. 1029–1034 (2004)
11. Thomas, U., Wahl, F.M., Maaß, J., Hesselbach, J.: Towards a new concept of robot programming in high speed assembly applications. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Canada, pp. 3827–3833 (2005)
12. Maaß, J., Molkenstruck, S., Thomas, U., Hesselbach, J., Wahl, F.M.: Definition and execution of a generic assembly programming paradigm. Assembly Automation 28(1), 61–68 (2008)
13. Lieberman, L.I., Wesley, M.A.: AUTOPASS: An automatic programming system for computer controlled mechanical assembly. IBM Journal of Rersearch and Development 21(4), 321–333 (1977)
14. Lozano-Pérez, T., Winston, P.: LAMA: A language for automatic assembly. In: Proc. of the International Joint Conferences on Artificial Intelligence, pp. 710–716 (1977)
15. Ambler, A.P., Popplestone, R.J.: Inferring the position of bodies from specified spatial relationships. Artificial Intelligence 6(2), 124–157 (1975)
16. Levi, P.: Planen für autonome Montageroboter (in German). Springer, Heidelberg (1988)
17. Bourjault, A.: Methodology of Assembly Automation: A New Approach. Springer, New York (1987)
18. Wilson, R.H.: On Geometric Assembly Planning. Ph.D. thesis, Department of Computer Science, Stanford University, Stanford, CA, USA (1992)
19. Halperin, D., Wilson, R.H.: Assembly partitioning along simple paths: The case of multiple translations. In: Proc. of the IEEE International Conference on Robotics and Automation, Nagoya, Japan, vol. 2, pp. 1585–1592 (1995)
20. Kaufman, S.G., Wilson, R.H., Jones, R.E., Calton, T.L., Ames, A.L.: The archimedes 2 mechanical assembly planning system. In: Proc. of the IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, vol. 4, pp. 3361–3368 (1996)

21. Knoll, A., Hildenbrandt, B., Zhang, J.: Instructing cooperating assembly robots through situated dialogues in natural languages. In: Proc. of the IEEE International Conference on Robotics and Automation, Albuquerque, NM, USA, vol. 1, pp. 888–894 (1997)
22. Schwarzer, F., Schweikard, A., Joskowicz, L.: Efficient linear unboundedness testing: Algorithm and applications to translational assembly planning. The International Journal of Robotics Research 19(9), 817–834 (2000)
23. Ostrovsky-Berman, Y., Joskowicz, L.: Relative position computation for assembly planning with planar toleranced parts. The International Journal of Robotics Research 25(2), 147–170 (2006)
24. Laugier, C.: Planning fine motion strategies by reasoning in the contact space. In: Proc. of the IEEE International Conference on Robotics and Automation, Scottsdale, AZ, USA, vol. 2, pp. 653–659 (1989)
25. Hirai, S.: Analysis and Planning of Manipulation Using the Theory of Polyhedral Convex Cones. Ph.D. thesis, Kyoto University, Kyoto, Japan (1991)
26. Xiao, J.: Goal-contact relaxation graphs for contact-based fine motion planning. In: Proc. of the IEEE International Symposium on Assembly and Task Planning, Marina del Rey, CA, USA, pp. 25–30 (1997)
27. Ji, X., Xiao, J.: Automatic generation of high-level contact state space. In: Proc. of the IEEE International Conference on Robotics and Automation, Detroit, MI, USA, vol. 1, pp. 238–244 (1999)
28. Lefebvre, T., Bruynincks, H., Schutter, J.D.: Polyhedral contact formation identification for autonomous compliant motion: Exact nonlinear bayesian filtering. IEEE Trans. on Robotics 21(1), 124–129 (2005)
29. Meeussen, W., Staffetti, E., Bruyninckx, H., Xiao, J., De Schutter, J.J.: Integration of planning and execution in force controlled compliant motion. Robotics and Autonomous Systems 56(5), 437–450 (2008)
30. Kavraki, L.E., Kolountzakis, M.N.: Partitioning a planar assembly into two connected parts is np-complete. Information Processing Letters 55(3), 159–165 (1995)
31. Lozano-Pérez, T.: Spatial planning: A configuration space approach. IEEE Trans. on Computers, C 32(2), 108–120 (1983)
32. Charzelle, B., Palios, L.: Decomposing the boundary of a nonconvex polyhedron. Algorithmica 17(3), 245–265 (1997)
33. Thomas, U.: Automatisierte Programmierung von Robotern für Montageaufgaben (in German). Shaker Verlag, Aachen (2008)
34. Scaramelli, L., Mello, H.D.: Task Sequence Planning for Robotic Assembly. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA (1989)
35. Maaß, J., Reisinger, T., Hesselbach, J., Schumacher, W.: A versatile robot control architecture for sensor integrated assembly. In: Proc. of ISR/Robotik (2006)
36. Kröger, T., Finkemeyer, B., Wahl, F.M.: Manipulation Primitives — A Universal Interface Between Sensor-Based Motion Control and Robot Programming. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 293–313. Springer, Heidelberg (2010)
37. Maaß, J., Dietrich, F., Hesselbach, J.: RCA562: Control Architecture for Parallel Kinematic Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 315–331. Springer, Heidelberg (2010)
38. Lefebvre, T., Bruynincks, H., Schutter, J.D.: Polyhedral Contact Formation Modelling and Indentification for Autonomous Compliant Motion. IEEE Trans. on Robotics and Automation 19(1), 26–41 (2003)

Self-management within a Software Architecture for Parallel Kinematic Machines

Jens Steiner, Ursula Goltz, and Jochen Maaß

Abstract. This paper presents a software architecture for the control of parallel kinematic machines. A generic approach for the integration of self-management capabilities is introduced and detailed with the example of a management component for the dynamic distribution of tasks. The goal is to optimize the load balance of the distributed control system while minimizing bus communication. Depending on monitored information, possible system adaptations are calculated, which are verified regarding their conformance to real-time requirements before they are stored in a data base. A combination of graph partitioning heuristics is used to determine nearly optimal control task to CPU mappings.

1 Introduction

To handle the ever growing complexity of software controlled systems has become a major issue over the last few years. Not only the size of systems is growing but also the connectivity of possibly heterogeneous subcomponents. Accordingly, problems regarding maintainability and extensibility increase, and intricate errors lead to unpredicted and in cases catastrophic behavior. In the future there might be systems of a complexity that exceeds full human comprehension and still need to be maintained or extended. For this reason, an active research field today are systems that can adapt themselves to changing external stimuli, without or with just little user interaction, and help human operators to gain control of the complexity.

Jens Steiner · Ursula Goltz
Technische Universität Braunschweig, Institute for Programming and Reactive Systems, Mühlenpfordtstraße 23, 38106 Braunschweig, Germany
e-mail: {steiner,goltz}@ips.cs.tu-bs.de

Jochen Maaß
Skysails GmbH & Co. KG, Veritaskai 3, 21079 Hamburg, Germany
e-mail: j.maass@tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 355–371.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

The Autonomic Computing initiative started by IBM in 2001 proposed systems that, e.g., strive for optimization of their own behavior, recognize, predict, and handle errors, configure and reconfigure themselves [1]. They should realize so called self-x or self-* properties like self-configuration, self-healing, self-optimization, and self-protection [2]. A similar idea are organic computing systems, which also try to handle complexity with concepts inspired by nature [3]. Self-management is a concept that all autonomic, adaptive or organic computing systems have in common. Its goal is not only to gain control of complexity but also to create robust and fault-tolerant systems, which can adapt to changing conditions or even protect themselves from environmental hazards.

To integrate such self-management capabilities into dependable systems with, e.g., safety or real-time constraints, poses many new challenges [4]. If safety critical software systems are to be extended by self-management features or are just to be connected to partly autonomous components, safety properties that were existent before have to be retained by all means. The same applies to real-time critical systems where hard deadlines exist and late answers are as bad as wrong ones.

The collaborative research center 562 (SFB 562) investigates methodical and component related fundamentals for the development of robot systems based on closed kinematic chains [5]. Such Parallel Kinematic Machines (PKMs) are usually constructed with passive structure limbs and have their drives mounted on a stationary platform. Due to low moved masses, PKMs achieve very high velocities and accelerations. Moreover, they have a high structural stiffness and repeating accuracy and excel in handling and assembly tasks. One disadvantage of PKMs is their small workspace to installation space ratio compared to serial robots. Regarding its task setting, a single PKM is not as flexible as a serial robot. Due to this, a large variety of different structures exists, all constructed and optimized for a special purpose. A generic software architecture has been developed in the SFB 562, which is called *PROSA-X* (**P**arallel **RO**bots **S**oftware **A**rchitecture - e**X**tended) and can be used to control different types of PKMs. Fig. 1 shows two of the robots developed in the SFB 562, HEXA and TRIGLIDE. They are both controlled by the same software with differences only in small robot specific modules.

Fig. 1. Demonstrators HEXA and TRIGLIDE

The high velocities and accelerations reached by PKMs induce special requirements for the control software running on off-the-shelf PCs. Since any software malfunction can cause the destruction of the robot, the whole system is safety critical [6]. The high control frequencies needed for the control of PKMs induce several real-time requirements for the system. Some of the deadlines for the periodically executed control tasks are hard and if violated can endanger the robot structure. Although there were high requirements regarding safety and real-time, self-management capabilities have been integrated in the system. Self-optimization and self-healing capabilities are provided by a management component that analyzes the current task setting and system topology and calculates nearly optimal control task to CPU resource mappings. The new distributed control system seamlessly integrates CPU resources that become available and reconfigures itself after topology changes.

The remainder of the article is structured as follows. Sec. 2 describes the control system that is self-configuring and rearranges itself according to the actual robot task. Sec. 3 introduces a generic design for management components, which has been used to integrate further self-* properties into the robot control system. Sec. 4 details how any system adaptation is verified before it is applied. The focus here are the real-time requirements of the system that have to be ensured. Sec. 5 concludes the article.

2 Control Architecture

In this section, the robot control architecture of *PROSA-X* and its design principles are described. As a key technology it incorporates the flexible and highly efficient communication and synchronization middleware *MiRPA-X* [7, 8]. It provides similar functionality as its predecessor *MiRPA* [9, 10] but differs in design, implementation and performance. *MiRPA-X* uses message passing and other features of the real-time operating system QNX for the realization of synchronous and asynchronous communication services. Due to this, the control architecture can be ported to any real-time operation system supporting similar mechanisms. Application processes providing control level services are regarded as servers, while service requesters are regarded as clients. Servers are blocked until they receive specific queries and instructions, clients block when they request functionality. Alongside the message-based communication, *MiRPA-X* provides a communication mechanism based on shared memory usage. As for messages, the shared memory mechanism uses the *MiRPA-X* name service. The usage of shared memory facilitates a high-speed data transfer between application processes, without the object server being involved in the actual communication task. After registration, application processes directly read and write inter-process data using memory pointers provided by the middleware. To ensure the data integrity of those shared memory areas *MiRPA-X* provides synchronization mechanisms like mutexes and condition variables.

2.1 Layered Design

Fig. 2 shows the three layers of the architecture, the very right section representing the hard real-time layer with the shortest deadlines. Processes here operate in a token-passing context in a strictly serial and deterministic manner using a high cycle frequency (1-8 KHz). The processes in the token cycle have the highest priority in the system and are never preempted by other processes. Their execution cycle is triggered by a hardware clock generated interrupt from the IEEE 1394 link layer. This leads to an extremely small jitter and significant benefits for the subordinated drive controller [11]. The first process in the cycle (*IAP - Industrial Automation Protocol*) realizes the connection to the field bus [7]. It catches the cycle start interrupt and transmits the sensor data transferred via the IEEE 1394 bus into a shared memory area. After the *IAP* has released its token, it is passed to the *Hardware Monitoring & Control* process. This process is responsible for activation and shutdown sequences of the robot, along with monitoring and surveillance functionality. Moreover, it activates the control core in the adjacent layer according to an adjustable ratio of cycles. The token is subsequently passed to the processes that encapsulate robot-specific controllers: The *Drive Controller* and the *Smart Material Controller*, which is applied to reduce vibrations induced by high-speed motion [12]. Finally, the token returns to the *IAP* process and the output values are transmitted to the actuators of the robot.

While the *IAP* is waiting for the next cycle start interrupt, processes in the message-based layer of the architecture are executed. This middle layer is responsible for generating a Cartesian space trajectory with respect to the actual skill primitive that is executed. A Cartesian trajectory is required, since it provides abstraction from a particular robot by defining the motion of the end-effector in distinction from

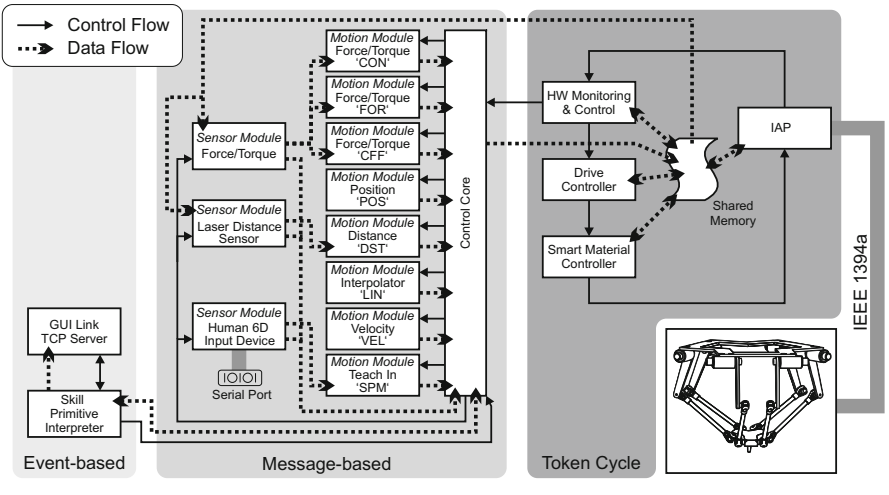


Fig. 2. Information flow in different layers of the control architecture

defining the motion of the actuated joints. The layer uses message passing for communication between processes. When a synchronization event from the Token Cycle layer occurs, the sensor module processes are notified by a multicast message in order to read data from the shared memory where the sensor data is stored. The sensor modules encapsulate signal processing algorithms, such as filtering or coordinate transformations into the task frame and pass the processed information to the motion module processes. The motion modules encapsulate the trajectory generation algorithms and are explained in detail in the next subsection. They are activated by the control core by a point-to-point message using the highly efficient name-service of the middleware only if required for the execution of the actual skill primitive (see Sec. 2.2). Detailed information on control engineering aspects is provided in [11]. After a configurable number of token cycles has passed, the *Hardware Monitoring & Control* process synchronizes the control core again and the data from the motion modules is fusioned to a valid set of Cartesian trajectory data. This information is passed to the *Drive Controller* via the shared memory area located in the Token Cycle layer and protected by a mutex.

The real-time requirements for the middle and right layer are hard. Missed deadlines are noticed by the control system and lead to a controlled shutdown. In contrast to this, the Event-based layer is not real-time constrained and executes the robot program interpreter and the GUI link server.

2.2 Motion Modules and Robot Programming

The motion modules, being the most important processes in trajectory generation, are subject to encapsulation of a large variety of task-optimized control algorithms. Using them, the control engineer can focus on the robot task, without the need for expert knowledge regarding the specific robot or communication and synchronization issues.

The base class of a motion module features an abstract interface to the motion algorithm and to a profile information class. The only job a programmer has to perform, apart from implementing the algorithm, is to fill in the profile. The profile contains the name by which the control algorithm is specified in the controller matrix of the skill primitive (compare Figure 3). Furthermore, the sensor data required to carry out the algorithm is specified by name. This allows the self-configuring control core to establish the communication links automatically. Additionally, the maximum and minimum cycle frequency the algorithm has been designed for, as well as the experimentally determined worst case execution time, is required. This information is used for an analysis of the real-time requirements described in Sec. 4.

The control system of *PROSA-X* uses a programming interface based on skill primitives [13]. It allows to program the robot on a task level and focuses on the description of a manipulation motion in a Cartesian task frame that can be placed arbitrarily. Fig. 3 shows an example of a skill primitive where a T-shaped block is in contact with a bent surface that it is to follow while maintaining the contact. This single skill primitive is just a part of a more complex manipulation task that

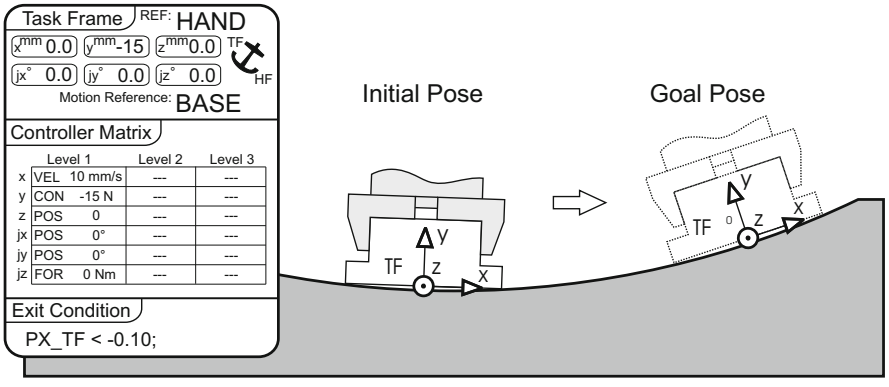


Fig. 3. Skill primitive example

is defined in a skill primitive net. The skill primitive in Fig. 3 demonstrates some specifics of the skill primitive concept that are relevant for the scheduling analysis described in Sec. 4. A more detailed view on skill primitives, respectively manipulation primitives, can be found in [14, 15, 16, 17].

The rows of the controller matrix represent the degrees of freedom of the robot. The columns define the level of the controller. Here, only the top level is used, but succeeding levels can contain alternative controllers that become active when preceeding ones can not operate, e.g., because of a missing sensor signal. Each entry in the control matrix specifies a control algorithm and the according desired value. The example shows four active controllers (*VEL, CON, POS, FOR*) with the task frame placed at the bottom of the block (y -15). The rotation of the task frame around the z-axis is controlled by the force controller *FOR*. The desired value for the torque is zero, so the block stays aligned to the contour. In this skill primitive, the task frame is attached to the hand frame and the velocity controller *VEL* moves the block alongside the contour. In y-direction *CON* is active, which is a combined force and velocity controller and maintains the contact. The exit condition is specified to end the skill primitive when the block has moved a certain distance along the contour.

3 Self-management

Plans to integrate vision-based control in the system to realize, e.g., camera based bin picking or visual servoing tasks have lead to a distributed version of *PROSA-X* where multiple control PCs can share the algorithmic load. This is also a prerequisite for an online use of singularity avoiding path planning algorithms for the HEXA robot [18]. On a single control PC those algorithms can only be used offline (while there is no movement of the end effector) because in combination with other control tasks they would violate real-time requirements. Fig. 4 shows the distributed version

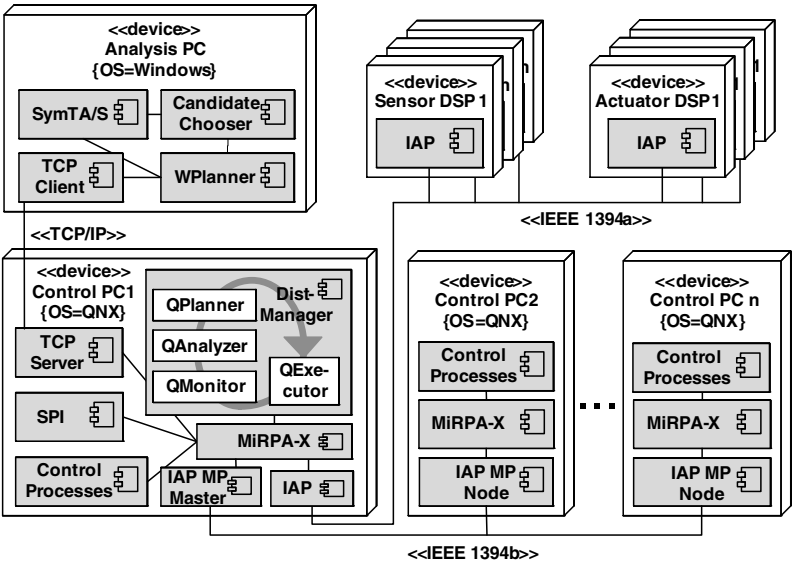


Fig. 4. Deployment diagram of distributed PROSA-X

of *PROSA-X* with control processes, an instance of the middleware *MiRPA-X*, and a variant of the FireWire bus protocol *IAP* running on each of the control PCs.

The *Cycle Start Telegram* (CST) of the first FireWire bus (IEEE 1394a) connecting the sensor and actuator DSPs is used for a system wide clock synchronization and as a periodic trigger for the robot control. The additional control PCs connected via the second FireWire bus (IEEE 1394b) communicate demand oriented and can not cause disturbances on the critical first bus. *SPI* represents the interpreter for the skill primitive net [14] containing the robot program. The component *DistManager* running on *Control PC 1* is detailed in Sec. 3.2. Why there is an a additional *Analysis PC* connected with a TCP/IP connection is explained in Sec. 4.

3.1 Management Components

A generic design pattern for management components to be integrated in *PROSA-X* has been proposed in [19]. Fig. 5 shows the pattern that follows the design of autonomic managers as introduced by IBM [1]. Likewise, a *PROSA-X* self-manager (e.g., *DistManager* in Fig. 4) has parts to record actual data (*Monitor*), analyze the current situation (*Analyzer*), plan an according reaction (*Planner*), and realize it accordingly (*Executor).*

Another class shown in Fig. 5 is *SelfManager*. Its constructor has a *SelfManagerPartFactory* as a parameter. A concrete factory class extends the abstract *SelfManagerPartFactory* and overrides its virtual creation functions. Such concrete part factory will create concrete part objects (like *QMonitor*, *QAnalyzer*, *QPlanner*, and *QExecutor* in Fig. 4). They are instances of classes that are derived from the abstract

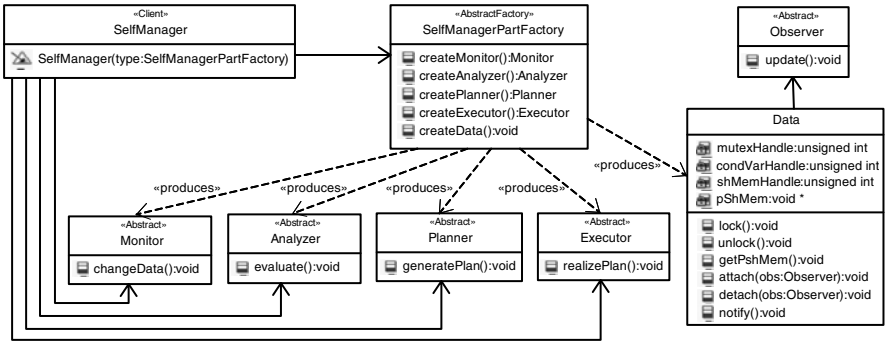


Fig. 5. Design pattern for PROSA-X self-managers

part classes (*Monitor*, *Analyzer*, *Planner*, *Executor*, and *Data* in Fig. 5). The design pattern used here is called *Abstract Factory* [20]. It is used to specify an interface to a family of related or dependent objects without specifying their concrete classes. This way, different self-managers can be realized without the need to change the *SelfManager* class. The interface to all its parts stays the same. *SelfManager* just calls the creation functions of the according factory object and triggers the behavior in its parts by calling the methods defined in the abstract part classes (*evaluate()*, *generatePlan()*, ...).

3.1.1 Performance Oriented Observer

Another design pattern that has been used to define the self-manager pattern is the *Observer* pattern [20]. The parts of *PROSA-X* self-managers communicate via data objects where they register as observers and get notified about changes. The realization of the observer pattern partly moved away from the object oriented design. By using QNX features like mutexes, condition variables and shared memory it was possible to reach a high-performance observer realization. This also has benefits for the integration of the self-managers since shared memory allows them to easily read system data relevant for evaluating the current situation and influence the system by providing a reaction also in a shared memory region. Furthermore, using the pattern realization shown in Fig. 6, the management component does not have to be a single process. Since all communication is realized by accessing the shared memory regions of the data objects it is possible that all the parts of a self-manager run in separate processes, or that there are for instance several monitor processes running in the system. The shared memory data objects also have the advantage that it is possible to modify only a small part of it, e.g., a single variable. For most of the access scenarios this is performing better than the original object-oriented observer pattern where the whole data object has to be written for every single access.

Fig. 6 also shows how the middleware *MiRPA-X* is used to realize inter-process communication and synchronization. Its API integrates QNX features (mutexes, condition variables, shared memory) and realizes an easy and standardized access to

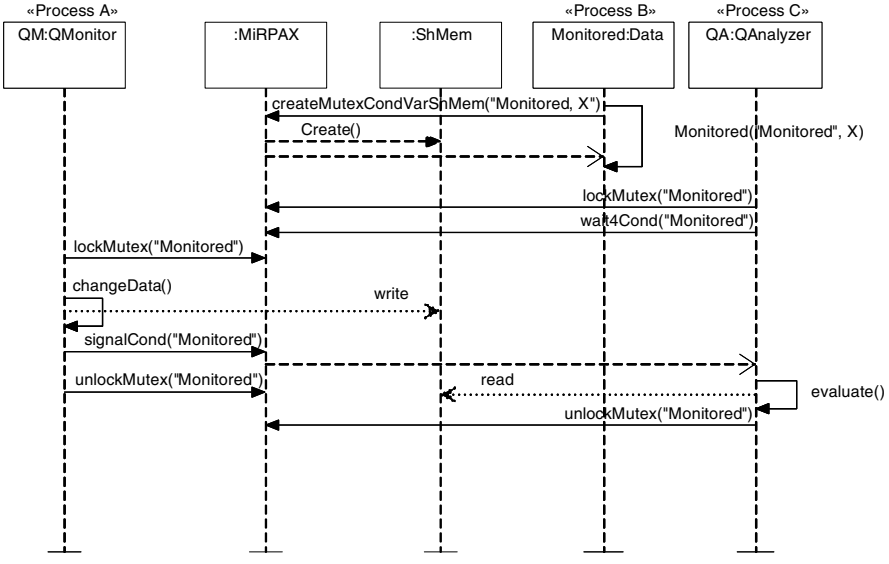


Fig. 6. Performance oriented observer realization

them. In Fig. 6 it is shown how the *Data* object *Monitored* creates a shared memory area of size X, a mutex, and a condition variable in its constructor. The monitor *QM* knows the name of the data object and via *MiRPA-X* can lock the according mutex, access the shared memory region, and signalize a change, even though it is running in a different process. *QA*, which is an observer of *Monitored* and runs in yet another process, locks the according mutex and waits for a change of the according condition variable (the wait call releases the mutex). After a change has been signalized, *QA* and other observers are unblocked and try to relock the mutex. They read and evaluate the data in the shared memory area sequentially.

3.2 Control Task Distribution

The integration of vision based control causes additional load for the communication system and is also a challenging task considering its inclusion in the scheduling of control tasks. In *PROSA-X* it only becomes possible with a distributed approach (see Fig. 4). Even then, a naive distribution of control tasks within this system may cause problems due to increased communication latencies and missed deadlines. With a huge number of possible task to CPU resource mappings (in the following called distribution pattern) and possibly changing task sets and topology, the question arises, what the best distribution pattern for a certain situation is.

As a response to this question, a self-manager for the automatic distribution of control tasks has been developed (*DistManager* in Fig. 4). It monitors system topology, task setting and execution times (*QMonitor*), analyzes the current situation (*QAnalyzer*), searches the big number of possible distribution patterns for a

candidate best fitting the current situation (*QPlanner*), and realizes its application (*QExecutor*). The cyclic execution of those steps realizes a control loop in which the system adaptation takes place. The direction of that adaptation is detailed in Sec. 3.2.1.

The result is a control system that automatically adapts to changing situations. It is capable to integrate and use additional control PCs. This is a self-optimization scenario because it allows for higher control frequencies and thus for a better quality of control. The control system also reacts upon the breakdown of a control PC by trying to generate a distribution pattern that can put the system back in operation. This is a self-healing scenario because any such breakdown will put the control system into an error state. Recovery from certain errors is not possible, though. No valid distribution pattern might be found and a breakdown of the main control PC can not be compensated. At the moment, it is a single point of failure. The control system also adapts according to changes in the current task setting. This task setting defines which parts of the control are currently in use (motion/sensor modules) and contains also more detailed information like data dependencies and execution times.

The control system operates under several hard real-time constraints, and deadlines exist, which cause the system to enter an error state if missed. Therefore, the planning of a system adaptation also comprises a scheduling analysis that ensures the adherence to all real-time requirements. Details about this pattern verification can be found in Sec. 4.

3.2.1 Calculating Distribution Patterns

The different parts of the control system have data dependencies which cause additional bus load and communication latencies when they are distributed to separated control PCs. This raises the question what a "good" distribution pattern is. One criterion certainly is communication latency since several real-time constrained execution paths exist in the system. Another criterion is balance. If the available CPU resources have a similar utilization and this utilization is low enough, there might be the chance to use higher control frequencies and to achieve a better quality of control.

The number of possibilities to distribute n control tasks to m resources is m^n. The effort necessary to find the optimal distribution pattern with perfect resource balance and minimal bus communication rises exponentially with a growing number of tasks or resources. This problem is known in parallel computing as task clustering problem and is of NP-hard complexity. Thus, it can not be solved in polynomial time and calculating the optimal pattern will be constrained to small systems.

However, there are heuristics that are fast in delivering nearly optimal solutions. The *DistManager* component uses a graph partitioning approach to generate potential distribution patterns [21]. Therefore, the *WPlanner* depicted in Fig. 4 generates a graph representation of the system where data dependencies become weighted edges and worst case execution times of periodically executed tasks become node weights. The resulting undirected node and edge weighted graph is partitioned in a way that the cut size is minimized and the resulting partitions are of equal or similar

size. The backward transformation delivers a distribution pattern with minimal bus load and CPU resources with similar utilization.

For the graph partitioning the Fiedler vector of the graph representation is numerically determined. This vector is the Eigenvector to the second smallest Eigenvalue of the graph Laplacian matrix which is also known as algebraic connectivity. The fields of the Fiedler vector hold information, how strongly connected the different graph nodes are, and thus can be used for a cut-minimizing partitioning. In addition to the Fiedler vector, an adaptation of the Kernighan-Lin heuristic [22] is used which is strongly dependent on good initial partitions. The Kernighan-Lin algorithm weights cut size higher than partition balance which is in line with the idea of a minimized bus communication due to the real-time requirements. Details of the combined partitioning algorithm, which is realized in the *CandidateChooser* component (see Fig. 4), can be found in [21]. On a 2.4 GHz Intel CPU it calculates nearly optimal distribution patterns in less than 100 milliseconds (number of tasks ≤ 100, number of partitions ≤ 10, medium connectivity).

4 Control Task Distribution

The distribution patterns described in Sec. 3.2 contain a mapping of control tasks to available CPU resources and thus a possible system adaptation. The approach to generate those patterns ensures that bus communication is minimized with a similar or equal CPU utilization. That alone does not mean that the real-time constraints of the system are met. The control system has several real-time requirements that have to be ensured to prevent dangers for the robot structure (see Sec. 4.2). For this reason, each generated pattern gets transformed into an input model for the scheduling analysis tool SymTA/S [23] which is used to verify whether the real-time requirements are met. This is done for each pattern before it is stored and can be used for a system adaptation. The model transformation, analysis, and back transformation is done on a dedicated PC running Microsoft Windows XP. The sequence diagram shown in Fig. 7 visualizes this process.

The *QPlanner* has been notified that one of the self-management scenarios has been initiated. It is not shown here how *QPlanner* gets notified by *QAnalyzer* and reads the actual system representation (see Sec. 4.1) out of the *Monitored* shared memory area. *QPlanner* first checks a hash table in the *Plans* shared memory to see if an adaptation strategy (distribution pattern) for the current situation is already available. If this is not the case a system model is transferred to the *WPlanner* component via a TCP/IP connection where it is transformed into an undirected edge and node weighted graph. The *CandidateChooser* component is used to partition the graph and an according input model for SymTA/S is generated. A SymTA/S server is used to calculate the worst case response times for all real-time critical execution paths in the system and the utilization of all CPU resources. If real-time requirements are violated, this process can be repeated several times with a parameter variation in the partitioning algorithm. If a distribution pattern satisfying all

constraints can be found this verified pattern is returned to the *QPlanner*, which stores the pattern and notifies the *QExec*utor.

4.1 System Representation

The system model that is used to generate the graphs for the partitioning algorithm and also the input models for the SymTA/S analysis contains information about all periodically activated parts of *PROSA-X*. The different software components are mapped to tasks to which different properties are assigned:

- worst and best case execution time
- quantified data dependencies to other tasks
- shared resources (shared memory, mutex, condition variable)
- assigned CPU or bus resource
- triggering source

Sources are triggers for a periodic activation and as such have a frequency. Fig. 8 shows two sources (*Source0, Source1*) with the same frequency of 1 kHz that activate tasks in the system with an offset of 750 μs. In PROSA-X, the *Cycle Start Telegram* (CST) of the IEEE 1394 bus is used for a system wide synchronization and synchronized task activation. It is generated every 125 μs. The example in Fig. 8 shows the system running with a 1 kHz control cycle, meaning that every first and

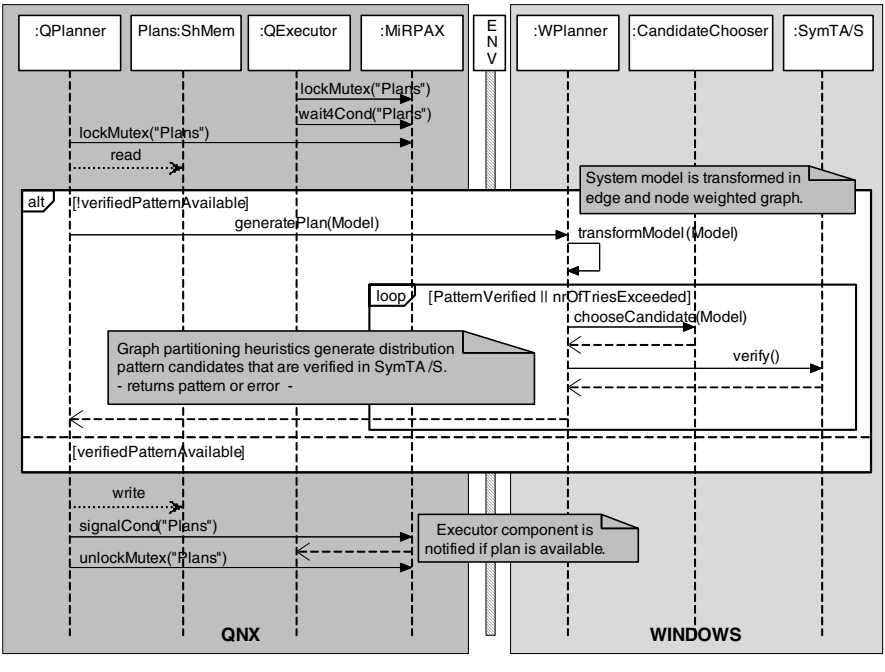


Fig. 7. Distribution pattern generation and verification

sixth CST activates a task chain. For CPU and bus resources the scheduling algorithm and performance are contained in the system model. Also denoted in it are all task chains with real-time requirements (see Sec. 4.2). For those, the start and end point as well as the deadline are defined.

The worst and best case execution times have been experimentally determined on the control PC. The execution times of control tasks are also monitored during the runtime of the system and, if necessary, are used to update the system model. An execution time beyond the worst case triggers a self-protection scenario where already verified distribution patterns are made invalid and are removed from the *Plans* data base.

In the original approach for building the initial system model, it was assumed that all motion and sensor modules are used simultaneously in the system [19, 21, 24]. As described in Sec. 2, different controllers might be specified for each degree of freedom of the robot and also cascading controller responsibilities might be defined. If for instance a force controller can not calculate new nominal values because of a missing force sensor signal, another controller (e.g., velocity) can fill in for it.

An analysis of skill primitive nets, describing different robot tasks, has shown that the case where all motion and sensor modules are active in the same control cycle is highly unlikely to happen because certain combinations of controllers are not meaningful. Thus, instead of assuming this as the worst case, now the system model is built and updated based on the actual robot task described in the skill primitive net. By traversing the skill primitive net it can be concluded what controllers can become active during its execution and possible combinations of simultaneously used control and sensor modules can be derived. Fig. 8 shows the system representation for the example skill primitive of Sec. 2. In this example, position (*POS*), velocity (*VEL*) and force (*FOR*, *CON*) controllers are used.

When a topology change is noticed by the *QMonitor*, e.g., an additional control PC, the system model gets updated automatically. It is assumed that CPUs are all of the same type and that additional ones get connected via a separate IEEE 1394b bus. Placeholders for IAP tasks and telegrams get created in the model with a worst case execution time of value zero. For distribution patterns that are analyzed regarding their real-time properties, those zeroes are automatically replaced by values that are linear dependent on the data dependencies between tasks in the model.

4.2 Real-Time Requirements

The execution paths in the system that are constrained by real-time requirements are depicted in Fig. 8. The first path starts with task *IAP_D*, which extracts the actual values from the *Device Data Telegrams* (*DDT*), which are sent by the DSP nodes.

The next task in that path is the *Hardware Monitoring & Control* (*HWM*), which prepares the received data for the control system and activates the *Central Control* (*CC*) and the *Drive Control* (*DC*) task, which generates new nominal values for the drives. The last task in that chain is the *Smart Material Control* (*SMC*), responsible to calculate nominal values for the active vibration suppression based on

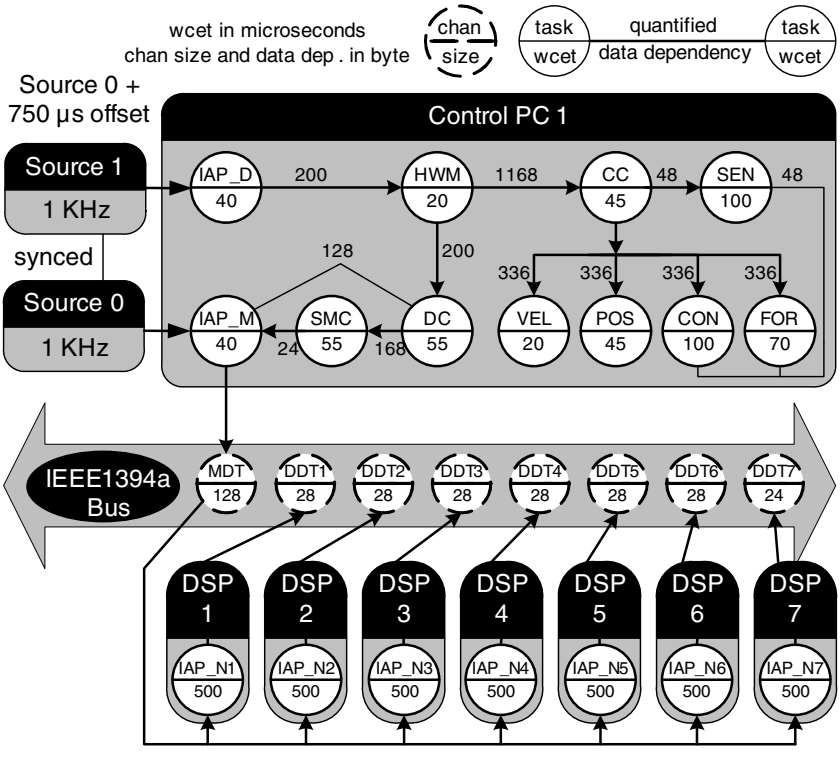


Fig. 8. Real-Time critical paths in PROSA-X

piezoelectric actuators [12]. The deadline for this task chain, which represents the rightmost layer in Fig. 2, is 250 μs after the activation of *IAP_D*. At this time the *IAP_M* task starts building the *Master Data Telegram* (*MDT*) containing the new nominal values.

From the time the *IAP_M* task gets activated and transfers the nominal values via the IEEE 1394a bus until all DSP nodes have responded with a new set of actual values in a DDT, a maximum of 750 μs may pass. This constraint results in a whole set of real-time critical paths: *IAP_M – DDTx*.

Another set of real-time constraint paths starts with the activation of the *Central Control* (*CC*) module. Depending on the actual robot task it activates necessary sensor and motion modules and prepares their results for a use by *DC*. Sensor and motion modules are executed with a lower control frequency than the subordinate controllers *DC* and *SMC* (here 1/2). The deadline for this set of paths is 1750 μs after the activation of *CC*. At that time it is checked if all activated motion modules have generated a result. If this is not the case, a braking trajectory is calculated, which stops the robot. This set of task chains represents the middle layer of Fig. 2.

The values for the deadlines are for a 1 kHz cycle frequency of the token cycle layer and a 500 Hz cycle frequency of the message-based layer in Fig. 2.

5 Conclusion

This paper introduces the generic software architecture *PROSA-X* fulfilling the special requirements of parallel kinematic machines. It is used to control several parallel robots with differences only in a small robot specific part. *PROSA-X* contains several parts that have been designed to be self-configuring. Depending on the actual robot task and the motion and sensor modules that have registered functionality via the middleware *MiRPA-X*, the control core decides whether it can execute a certain task and automatically establishes connections to the necessary modules.

The complex architecture was used as a case study for the integration of advanced self-management features into a dependable system. A design pattern for *PROSA-X* management components was introduced that comprises a performance oriented realization of the observer pattern. It uses *MiRPA-X* and QNX functionality to realize fast and safe inter-process communication and synchronization. Following the pattern, a management component has been realized, which is capable to adapt the distributed control system to changes in topology and task setting. It uses a graph partitioning approach to determine beneficial distribution patterns, which map control tasks to CPU resources while minimizing bus communication.

In the approach presented here, it is not only ensured that real-time requirements are met before and after the integration of a management component, but the according verification is integrated in the management component itself. An online use of the scheduling analysis tool SymTA/S guarantees that after each adaptation the system still fulfills its real-time requirements. Furthermore, errors in the model the scheduling analysis is based on, e.g., wrong worst case execution times, can be recognized and thus form an improvement compared to an offline analysis approach.

Acknowledgements. The authors would like to thank the DFG (German Research Foundation) for the financial support of the collaborative research center 562 "Robotic Systems for Handling and Assembly", and QNX Software Systems as well as the Symtavision GmbH for the grant of free software licenses.

References

1. Kephart, J.O., Chess, D.M.: The Vision of Autonomic Computing. Computer 36(1), 41–50 (2003)
2. Wolf, T.D., Holvoet, T.: A Taxonomy for Self-* Properties in Decentralised Autonomic Computing. In: Autonomic Computing: Concepts, Infrastructure, and Applications, Taylor & Francis, CRC Press (2006)
3. Branke, J., Mnif, M., Müller-Schloer, C., Prothmann, H., Richter, U., Rochner, F., Schmeck, H.: Organic computing – addressing complexity by controlled self-organization. In: Proceedings of the 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation, ISoLA 2006 (2006)

4. Frank, U., Giese, H., Müller, T., Oberthür, S., Romaus, C., Tichy, M., Vöcking, H.: Potenziale und Risiken der Selbstoptimierung für die Verlässlichkeit mechatronischer Systeme. In: 5. Paderborner Workshop Entwurf mechatronischer Systeme. HNI Verlagsschriftenreihe, Paderborn, vol. 210 (2007)
5. Merlet, J.-P.: Parallel Robots. Kluwer Academic Publishers, Dordrecht (2000)
6. Storey, N.: Safety-Critical Computer Systems. Pearson, London (1996)
7. Kohn, N., Varchmin, J.-U., Steiner, J., Goltz, U.: Universal communication architecture for high-dynamic robot systems using qnx. In: Proceedings of International Conference on Control, Automation, Robotics and Vision (ICARCV 8th), vol. 1, pp. 205–210. IEEE Computer Society, Kunming (2004)
8. Dadji, Y., Michalik, H., Kohn, N., Steiner, J., Beckmann, G., Möglich, T., Varchmin, J.U.: A Communication Architecture for Distributed Real-Time Robot Control. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 213–231. Springer, Heidelberg (2010)
9. Finkemeyer, B.: Robotersteuerungsarchitektur auf der Basis von Aktionsprimitiven. Ph.D. thesis, Technische Universität Carolo-Wilhelmina zu Braunschweig (2004)
10. Finkemeyer, B., Kröger, T., Wahl, F.M.: A Middleware for High-Speed Distributed Real-Time Robotic Applications. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 193–212. Springer, Heidelberg (2010)
11. Maaß, J., Kohn, N., Hesselbach, J.: Open modular robot control architecture for assembly using the task frame formalism. International Journal of Advanced Robotic Systems 3(1), 1–10 (2006)
12. Keimer, R., Sinapius, M.: Adaptive components for parallel robots. In: Schütz, D., Raatz, A., Wahl, F.M. (eds.) Proceedings of 3rd International Colloquium of the Collaborative Research Center 562. Fortschritte in der Robotik, vol. 14, pp. 181–190. Shaker Verlag, Aachen (2008)
13. Thomas, U., Wahl, F.M.: Sensor guided execution of robot tasks based on skill primitives. In: Robotic 2002, pp. 71–77 (2002)
14. Thomas, U., Maaß, J., Hesselbach, J., Wahl, F.M.: Towards a new concept of robot programming in high speed assembly applications. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2005, pp. 3932–3938 (2005)
15. Maaß, J.: Ein Beitrag zur Steuerungstechnik für parallelkinematische Roboter in der Montage. Ph.D. thesis, Technische Universität Carolo-Wilhelmina zu Braunschweig (2009)
16. Kolbus, M., Reisinger, T., Maaß, J.: Robot Control based on Skill Primitives. In: Robotics and Applications: Sixth IASTED International Conference Proceedings (2005)
17. Kröger, T., Finkemeyer, B., Wahl, F.M.: Manipulation Primitives: A Universal Interface Between Sensor-Based Motion Control and Robot Programming. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 293–313. Springer, Heidelberg (2010)
18. Bier, C., Maaß, J., Campos, A., Queiroz, E.: Direct singularity avoidance strategy for the hexa parallel robot. In: 18th International Congress of Mechanical Engineering (COBEM 2005), Ouro Preto, Brasilien (2005)
19. Steiner, J., Hagner, M., Goltz, U.: Runtime analysis and adaptation of a hard real-time robotic control system. Journal of Computers (JCP) 2(10), 18–27 (2007)
20. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995)
21. Steiner, J., Amado, A., Goltz, U., Hagner, M., Huhn, M.: Engineering self-management into a robot control system. In: Proceedings of 3rd International Colloquium of the Collaborative Research Center 562, pp. 113–125 (2008)

22. Kernighan, B., Lin, S.: An efficient heuristic procedure for partitioning graphs. Bell Systems Technical Journal 49(2), 291–308 (1970)
23. Henia, R., Hamann, A., Jersak, M., Racu, R., Richter, K., Ernst, R.: System level performance analysis - the symta/s approach. IEEE Proceedings Computers and Digital Techniques 152(2), 148–166 (2005)
24. Maaß, J., Steiner, J., Amado, A., Hesselbach, J., Huhn, M., Raatz, A.: Self-management in a control architecture for parallel kinematic robots. In: Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2008, Brooklyn, New York, USA (2008)

Model Based Quality Assurance for a Robotic Software Architecture

Jens Steiner, Karsten Diethers, Matthias Hagner, and Ursula Goltz

Abstract. This paper describes a model-based quality assurance approach in the context of a software architecture for parallel kinematic machines (PKMs). Due to high velocities PKMs are safety critical and cause hard real-time requirements for the according control system. In a joint effort mechanical engineers, electrical engineers, and computer scientists designed a software architecture for the special requirements of PKMs.

Here, it is explained what kind of models have been used to specify the structure, behavior, and requirements of the software under development and what validation and verification techniques have been used to improve the quality of the resulting software system.

1 Introduction

Parallel Kinematic Machines (PKMs) are a special kind of robots with closed kinematic chains [1]. Often, they have a lightweight structure because their drives are mounted on a base plate and, in contrast to serial robots, have not to be moved. PKMs have a high structural stiffness, precision, and repeating accuracy. Furthermore, they can reach very high velocities and accelerations. Due to this, the control software for PKMs is safety critical and has several hard real-time requirements.

Since PKMs lack flexibility compared to serial robots and have a smaller workspace to installation space ratio, they are usually constructed for a special purpose or application scenario.

Jens Steiner · Matthias Hagner · Ursula Goltz
Technische Universität Braunschweig, Institute for Programming and Reactive Systems, Mühlenpfordtstraße 23, 38106 Braunschweig, Germany
e-mail: {steiner,hagner,goltz}@ips.cs.tu-bs.de

Karsten Diethers
Capgemini sd&m AG, Carl-Wery-Straße 42, 81739 München, Germany
e-mail: karsten.diethers@capgemini-sdm.com

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 373–389.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

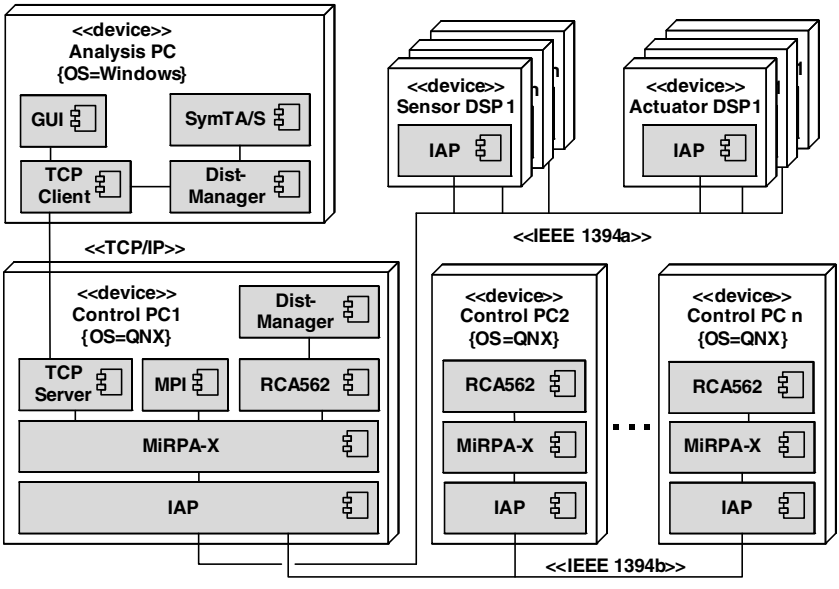


Fig. 1. PROSA-X deployment diagram

One important research activity within the SFB 562 was the development of a generic software architecture that could be used for different parallel robots. This software architecture was named *PROSA* (**P**arallel **Ro**bots **S**oftware **A**rchitecture) [2]. Fig. 1 shows a deployment diagram of the latest version of the architecture named *PROSA-X* (e**X**tended) that uses multiple control PCs to share the algorithmic load.

Since no commercially available communication system existed that met the rigid bandwidth and real-time requirements, a new one was developed in the SFB 562. It consists of a real-time capable middleware (*MiRPA-X*) and a bus protocol managing the communication via a bus system based on the IEEE 1394 standard (*IAP*). *MiRPA-X* provides similar functionality as its predecessor *MiRPA* [3, 4, 5] but differs in design, implementation and performance. Details about the whole communication system can be found in [6, 7].

A programming interface based on skill primitives, respectively manipulation primitives, has been realized that allows an intuitive and error tolerant definition of robot tasks [8, 9, 10]. The skill primitive nets can be specified in a *GUI* component on a Windows PC. Their interpreter (*MPI* in Fig. 1) is run on the main control PC under the real-time operating system QNX[1].

The control architecture *RCA562* [11] is shown in Fig. 1 with parts running on several control PCs. A management component (*DistManager*) adapts the distributed control system to changing external conditions like topology or robot

[1] http://www.qnx.com

task [12]. Via a TCP/IP connection, it uses the scheduling analysis tool SymTA/S[2] to verify the conformance of adaptations to real-time requirements [13] (see Sec. 4).

The development of *PROSA-X* was supported by an extensive model-based engineering approach. Several lightweight and formal quality assurance measures were applied based on models of the structure and behavior of the different software components. Validation, verification, and scheduling analysis techniques applied in different development stages made an early identification of design flaws possible and ensured a high quality end product conforming to safety and real-time requirements.

For the modeling of different aspects of *PROSA-X*, the Unified Modeling Language (UML) has been used. UML is a graphical modeling language for software intensive systems, which is standardized by the Object Management Group (OMG[3]). The possibility to reside on different abstraction levels and the amount of available diagram types made it the perfect choice for the interdisciplinary *PROSA-X* development process. A modeling tool that has been used from the beginning of the project was Rhapsody, which is now part of IBM's model-driven development environment for real-time or embedded systems engineering.

Sec. 2 describes a model-based validation approach based on Rhapsody's code generation capabilities. In Sec. 3, a model transformation from UML models into timed automata is introduced, which allowed a formal analysis of *PROSA-X* models. A scheduling analysis that guarantees the conformance of the system to its real-time requirements is detailed in Sec. 4.

2 Validation

The model-based development approach described here comprises a validation of system behavior already in the design phase [14]. After the static structure of *PROSA-X* had been specified using deployment diagrams and on a more detailed level class diagrams, interfaces were defined and communication relations to be realized via the middleware *MiRPA-X* were specified. Fig. 2 shows a class diagram example with classes of the control core and the skill primitive interpreter.

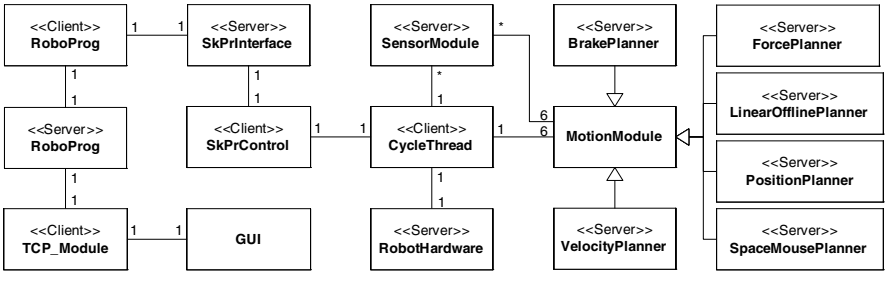


Fig. 2. Class diagram for control core and program interpreter

[2] http://www.symtavision.com

[3] http://www.omg.org

The middleware *MiRPA-X* allows only the role of a client or a server for one process. Servers can register functionality that can then be used by clients. Stereotypes have been used here to annotate the according roles. Control functionality is encapsulated in *MotionModules*, which are used by the *CycleThread* class according to the actual skill primitive that has been passed to the control system via the skill primitive interpreter and robot programming classes [11]. *MiRPA-X* does not only support message based communication but via its API offers a unified access to shared memory areas and according synchronization mechanisms. This is realized by using the according features of the real-time operating system QNX [6]. The *MiRPA-X* API has been defined early in the design process. It is used in all component models to specify their interactions in detail.

Based on this API is a simulation framework that has been developed to validate design decisions and the interoperability of software components for *PROSA-X,* based on UML models. This simulation framework provides most of the functionality of the middleware *MiRPA-X*. Since the correctness of *MiRPA-X* is essential for the operation of the system it has been analyzed with the verifying approach described in Sec. 3 [15]. Behavioral models of the middleware can not be validated in the simulation framework introduced here. Instead, UML state diagrams for all other *PROSA-X* threads have been created that use the aforementioned middleware API like a working black-box, e.g., on transitions. From within the used CASE tool Rhapsody, source code for the state diagrams is generated, which can be compiled and executed in the simulation framework. During this execution, calls to the middleware API functions are handled by the simulation engine. Thereto, each state diagram has to register itself in the simulation framework with its name, scheduling algorithm, priority, and client/server role. Afterwards, it can use functionalities for message passing (commands, requests), synchronization (mutex, sleepon locks, condition variables), and can access shared memory regions. The simulation backend also contains a virtual scheduler that maintains ready queues and activates/preempts the thread models according to their priority and scheduling algorithm. This scheduler is an exact copy of the QNX scheduler and also features priority inheritance to avoid priority inversion problems during the simulation. The simulated thread models behave like threads running in QNX and can enter most of the execution states of a real QNX thread (see Fig. 3). When they are not ready or running, they can be blocked while waiting for a message, shared resource or interrupt.

It is possible to influence the simulation by generating user input or virtual interrupts or by manipulating shared memory regions. During the simulation, all threads, their connections to each other, and their execution states can be visualized, as can all synchronization mechanisms and the content of shared memory areas. By using code instrumentation, Rhapsody is capable to visualize state diagrams during the simulation. This, together with the information shown in the simulation frontend, greatly eases the validation process.

Due to the inherent complexity of *PROSA-X,* with multitudes of components developed rather independently in several sub-projects of the SFB 562 and still exhibiting a lot of dependencies, it was essential to validate the behavior of the system

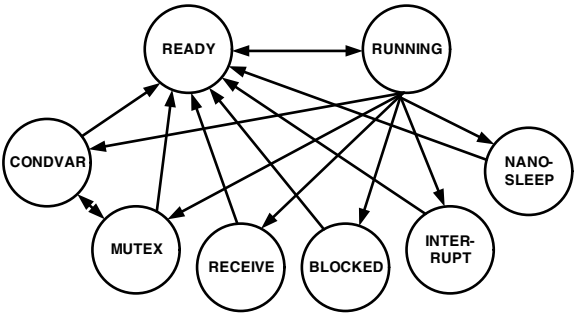


Fig. 3. Thread states supported by simulation framework

early in the design process. The simulation framework follows the idea of an executable specification [16], which is supported by the code generation capabilities of the used CASE tool Rhapsody. A validation of system behavior could be performed before the implementation of *PROSA-X* components, and several design flaws (livelocks, inconsistent interfaces) have been identified with the simulation framework. Later, the effects of a self-management component within *PROSA-X*, responsible for a dynamic redistribution of control tasks [13], have been validated in the simulation framework.

3 Verification

Another quality assurance measure applied in the SFB 562 is the verifying approach described in this subsection. Model checking techniques have been used to prove the correctness of the system model regarding a set of behavioral requirements [17, 18, 19]. Model checking is a formal verification technique, which is based on state space exploration. It allows to check a formal system model for unwanted conditions like deadlocks, livelocks, and error states, or invariants like "*request α will always be answered*". The modeling of *PROSA-X* components' behavior and requirements has been done with UML state diagrams and sequence diagrams. An automated model transformation has been realized that transforms the according UML models into input for the model checker UPPAAL [20].

The UML as a general purpose modeling language contains a certain set of diagram types that for some domains lack means to express domain specific information and constraints. For this reason it can be extended by profiles, which make special stereotypes and tagged values available to be used in models. The OMG itself standardized several such profiles, e.g., the *Profile for Schedulability, Performance and Time* (SPT), which provides ways to specify timing relevant information such as resources, scheduling parameters, tasks, and deadlines directly within UML models. A more recent replacement of the SPT is the *Profile for Modeling and Analysis of Real-time and Embedded Systems* (MARTE [21]). When the SFB 562 started, neither SPT nor MARTE were available yet; so, early work within the project did

use standard UML notation to describe timing constraints (e.g., labels attached to beginnings or ends of message arrows in sequence diagrams).

3.1 Sequence Diagrams

Since sequence diagrams do not deal with the implementation of instances but are well-suited for defining sequences of events, they can be used to model requirements within a system of communicating instances. For example, they can be used to specify that the response of a certain message will occur before a deadline is missed. Labels at message arrows are interpreted as time stamps here and have been used in timing constraints, e.g., to specify the minimum or maximum time gap between two marked points in the diagram or to define the duration of a periodic sequence. The sequence diagram on the left-hand side in Fig. 4 shows an example for this where less than 3 time units may pass between *t* and *u*.

With combined fragments it is possible to describe looping behavior in sequence diagrams. To deal with different occurrences of labeled events in loops, the following convention is introduced: *afirst* is used in constraints to refer to the first occurrence time of an event with time stamp *a* in the loop, *alast* refers to the last occurrence of the tagged event in the loop, and *anext* denotes the time of the event occurrence in the following iteration. Fig. 4 shows an example of this notation in the the sequence diagram on the right-hand side. Between the first occurrence of *t* and the last occurrence of *u* less than 10 time units may pass. The loop may take place between 1 and 5 times, which is denoted directly after the loop keyword. During the design process, the requirement status of a sequence diagram has to be declared. Thus, it has to be indicated whether it describes mandatory or optional behavior. As introduced here, they describe mandatory behavior (event sequences and deadlines).

3.2 State Diagrams

For modeling the behavior of systems, the UML provides state diagrams. Concerning the formal analysis of UML state diagrams with time, only that subset of

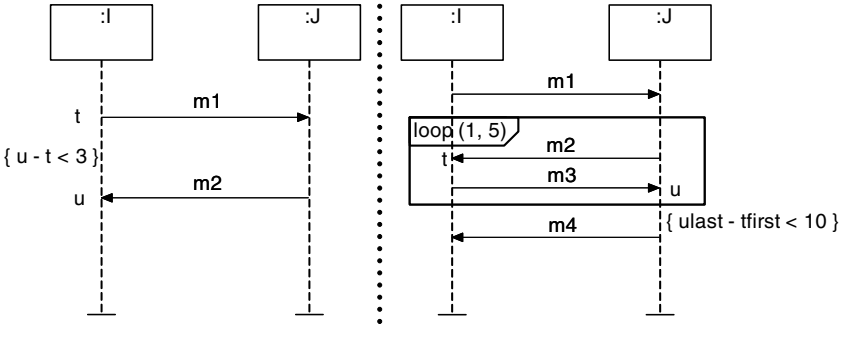


Fig. 4. Labels in sequence diagrams

syntactic elements is considered here that is used for the translation into timed automata. The most important restrictions are:

- Concurrent states within a state diagram are omitted to avoid overhead in implementation and verification of state diagrams. This restriction is adequate because here state diagrams model the behavior of objects that are implemented as threads without intra-object concurrency. Parallelism occurs only on top of objects. I.e. the family of state diagrams building the system runs concurrently.
- Each *action* that may occur as transition label has to have a simple arithmetic form, which can be directly transferred to UPPAAL to keep the verification decidable.
- Composite transitions, history connectors, entry/exit actions, and do activities of states are also omitted. These state diagram elements can be handled by an extension of the translation scheme.

UML state diagrams consist of states and transitions. A state can be *basic*, *final,* or *composite*. If a state is *basic* it has no substates. A *composite* state has at least one substate and exactly one of them is *active* at a certain point of time. *Composite* and *basic* states form a tree structure with a top-level *composite* state as root. This hierarchy allows for a stepwise refinement of the behavior of complex systems. Each *composite* state contains exactly one substate marked as *initial* and at most one *final* state. The *initial* state is the default entry point of the *composite* state. If a *composite* state is in a *final* state it can be left by a completion transition, i.e. a transition that is not triggered by an explicit event. States are connected by transitions. Only transitions having a unique source and target are handled here. Fig. 5 shows a simple example of a state diagram with hierarchy. The transition from *basic* state *C* to the *final* state of the top-level *composite* state visualizes the syntax of transition labels.

At the beginning of a state diagram execution, the root state and all recursively reachable initial substates are marked active. A transition is *enabled* if the source of the transition is an active state, the trigger event *e* occurs and the guard *g* evaluates to *true*. The transition will be taken immediately if there are no conflicting transitions in the set of enabled transitions. If transitions are in conflict, their source states are

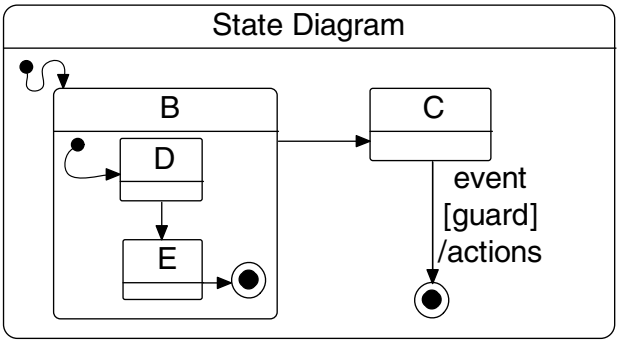


Fig. 5. Hierarchical state diagram

identical or related hierarchically. In the latter case, the UML priority scheme gives priority to transitions having source states lower in the hierarchy, i.e. only those can be chosen for firing. Otherwise, a transition is chosen non-deterministically. If a transition fires the list of actions a is performed. Moreover, the states that were active so far are marked inactive. The target state of the firing transition and all its parents and default initial substates become active.

In the setting described here, an event can be either a simple event, a completion event, or a time triggered event *after(min, max)* (relative to the activation time of the source). A guard is a Boolean expression and an action is either a sending of events to another state diagram or an assignment to variables. Sophisticated functions like the calculation of a trajectory of the robot are not supported. However, to verify the timing behavior of the software it is sufficient to map such calculations on their effect on time. The verifier does not have to describe the behavior in all detail but instead only models the timing effect by giving upper and lower bounds of the calculation time in *after*-expressions.

The semantics of a system of state diagrams defines event queues in each single state diagram, in which other state diagrams can insert events (action at a transition). Time elapses only if all event queues are empty. In this state the system is stable. If a timing event occurs, a step in the system is performed. A single transition fires and the effect of the transition is executed. Each state diagram can send events to arbitrary other state diagrams. Events are enqueued in the related event queue. A state diagram with a non-empty event queue processes an event without a time delay. If several state diagrams have non-empty event queues, the next active state diagram will be chosen non-deterministically. Selecting and processing an event will be repeated without a loss of time until all queues are empty. Then the system is in a stable state again until a timing event starts a new sequence of steps.

3.3 Model Transformation

For the verification, an appropriate formalism is required that allows an automatic model checking. Timed automata introduced by Alur and Dill in [22] are such a formalism, which extends the well-known notion of finite automata by clocks. In the following, it is referred to the variant of timed automata that is supported by the UPPAAL tool [20], allowing for efficient model checking of medium sized industrial case studies.

In timed automata, transitions between states are of the form $s \xrightarrow{g,\alpha,r} s'$ where g is a guard over clocks or integer variables, α is a synchronization action, and r is a set of assignments (resets) on clocks and variables. Synchronization is done via channels from a set A using send and receive actions $\mathcal{A} = \{a?, a! \mid a \in A\}$.

Interacting components can be modeled as networks of timed automata. Communication is possible on shared memory via global variables or by complementary labeled transitions, which have to be taken simultaneously (synchronous communication). Transitions are assumed to take no time. The amount of time elapsing in a state depends on its invariant, which defines a hard deadline for taking a transition.

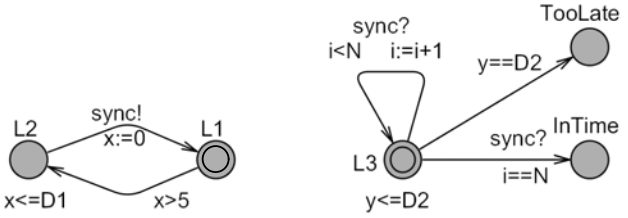


Fig. 6. Network of timed automata

Fig. 6 shows a basic example of a network of timed automata. More details can be found in [20].

Although state diagrams and timed automata both describe systems as structures consisting of states and triggered transitions labeled with guards and actions, there are fundamental differences that have to be taken into account if one formalism is translated into the other:

- Timed automata in UPPAAL lack hierarchy. When this model transformation was realized, there was no priority scheme for transitions in timed automata.
- A step within state diagrams is maximal and all transitions selected for a step fire simultaneously. In UPPAAL timed automata, transitions that are enabled do not have to proceed and usually fire sequentially.
- Communication by state diagram events is asynchronous. Communication via channels in UPPAAL timed automata is synchronous. A timed automaton that implements a queue is used here to simulate communication via events.

The model transformation from UML state diagrams to timed automata is divided into three steps: First all composed states of the state hierarchy are removed because timed automata in UPPAAL lack hierarchical concepts. Secondly, the flat state diagrams are translated into a system of timed automata. In the last step, control automata are added to ensure that the timed automata behave consistently to the state diagram semantics. For details regarding the transformation, the reader is referred to [19].

3.4 Analysis

For the analysis at UML level, a system description is needed in terms of the state diagrams as described in the previous chapter as well as a description of the desired behavior in terms of UML sequence diagrams. At the level of timed automata this maps to a network of timed automata for system description and a set of observer automata derived from the sequence diagrams. Both sets of automata are synchronized by timed automata actions (see [19] for details). For the analysis, the reachability check of the model checker UPPAAL is used. It is checked if the set of paths of desired actions is left and the observer automata is stuck in an error state [15, 18]. Automated checks are generated for the following system properties:

- The correct sequence of events, which is exchanged between the state diagram components. It is also checked if there are additional events after finishing the specified sequence.
- The correct performing of event loops according to loop constraints.
- The violation of deadlines. A violated timing constraint leads to a corresponding timing error state.

In the context of real-time applications, especially the third check is important. The combination of timing requirements and cyclic repetitions of event sequences leads to an explosion of the possible executions that cannot be checked manually, even for small systems. Therefore, automated checks are necessary.

4 Scheduling Analysis

The approach described in Sec. 3 uses sequence diagrams to annotate event orders and deadlines for communicating objects in the system. These constraints are verified within the model checker UPPAAL, which works fine for medium sized case studies. An advantage of this method is, that safety requirements can be verified (e.g., event orders, reachability of error states). However, for the analysis UPPAAL needs to build the state space of the system. This might fail for larger models due to memory consumption or time issues caused by the state explosion problem. Furthermore, the verification detailed in Sec. 3 only considers tasks running on the central control PC. It does not take into account communication via buses and tasks running on DSPs or additional PCs. And last, there is no notion of resources and their utilization. To overcome those issues, other ways to verify the real-time requirements of *PROSA-X* have been explored. In this section, a transformation from UML models into models for the scheduling analysis tool SymTA/S is described. SymTA/S uses a different analysis approach [23] that can handle all above mentioned problems.

A scheduling analysis in SymTA/S requires a system description containing relevant parameters: resources, tasks, channels and event streams. Resources can be processors or buses. Tasks encapsulate execution entities, which can be mapped to a processor, whereas channels encapsulate communication and can be mapped to a bus. Dependencies between those elements are described as event streams.

This kind of information exceeds the specification possibilities of standard UML. The UML profile MARTE [21] provides additional syntactical elements in terms of stereotypes and tagged values to describe system properties relevant for a scheduling analysis. The MARTE profile contains a lot more than what is described in this section, e.g., hardware properties can be described on a high level of detail. Here, only those MARTE elements used for the model transformation from UML models to SymTA/S input are introduced (see Table 1).

Some of the tagged values in the table are not used when the system is modeled. Instead, they can be used to annotate the models with results calculated during the scheduling analysis (*utilization* of CPUs or buses, *end2endT* - end to end response times of real-time critical paths, *isSched* - schedulable or not).

Table 1. Used stereotypes and tagged values of the MARTE Profile

Stereotype	Used on	Used Tagged Values
SaExecHost	Classes	*utilization, speedFactor*
SaCommHost	Classes	*utilization, speedFactor*
Scheduler	Classes	*schedPolicy, otherSchedPolicy, isPreemptible*
SchedulableResource	Classes	
SaStep	Methods	*priority, execTime*
SaCommStep	Methods	*priority, execTime, msgSize*
SaEnd2EndFlow	Activities	*end2endD, end2endT, isSched*
GaWorkloadEvent	AccepEventAction	*pattern*
Allocate	Associations	

4.1 Class Diagrams

Class diagrams have been used to specify the *PROSA-X* resources, tasks and their properties. CPUs and buses are modeled as classes annotated with the *SaExecHost* (CPU) or *SaCommHost* (bus) stereotype. They might show a *speedFactor* tagged value to emphasize performance differences and have a second tagged value *utilization* that can be used to annotate the resource utilization calculated during the scheduling analysis.

Threads or bus communication channels in *PROSA-X* are represented as classes with the *SchedulableResource* stereotype. They are connected to the host classes via dependencies (*Allocate*). The *SchedulableResource* classes can contain one ore more methods tagged with an *SaStep* stereotype. These methods represent tasks in the system that are performed by a thread depending on its activating input (see Sec. 4.2). *SchedulableResource* classes might also contain methods with an *SaCommStep* stereotype that represent communication tasks, i.e. telegrams sent via the bus

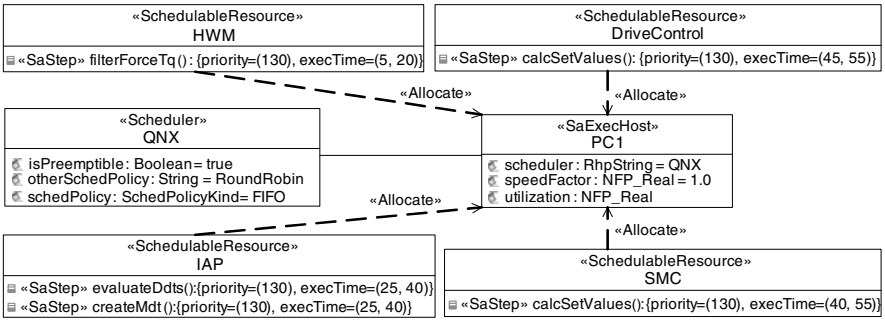


Fig. 7. Class diagram with MARTE annotations

resource the *SchedulableResource* has allocated. For the tasks or communication tasks, scheduling relevant information is annotated as tagged values (e.g., *execTime, priority*). Within the *execTime* tagged value, both, best and worst case execution times (measured on target hardware), are specified (see Fig. 7). For communication tasks, the amount of transferred data can be specified as *msgSize*.

No unit is specified for execution times and deadlines within the model. The person doing the annotations for the scheduling analysis has to consider this and has to ensure the same unit is used everywhere. Within the *PROSA-X* models, this implicit time unit is one microsecond. Fig. 7 shows a class diagram with MARTE annotations and classes representing the highest priority threads in *PROSA-X*.

4.2 Activity Diagrams

Elements of the MARTE profile have also been used within UML activity diagrams to describe real-time critical execution paths within *PROSA-X*. These activity diagrams are annotated with *SaEnd2EndFlow* stereotypes, which have an *end2endD* tagged value representing the deadline for the execution of the task chain. One activity diagram resembles a workload situation of the system and describes an execution order for tasks independently from the fact where the tasks are running. For the event, triggering the execution path, an arrival pattern is annotated. This trigger is represented in the activity diagram by a UML AccepEventAction, which serves as initial node and is annotated with the stereotype *GaWorkloadEvent.* For this stereotype a tagged value with the name *pattern* exists, which is used to define properties of the arrival pattern (e.g., periodic or sporadic).

Fig. 8 shows the properties of the *DDT* event as a comment since they are not visualized within Rhapsody. They will be exported, though, when the according model is written into an XMI file (see Sec. 4.3).

Within this figure, the periodic *DDT* event activates the task chain with the shortest deadline in *PROSA-X*. *DDT* stands for the arrival of the last *Device Data*

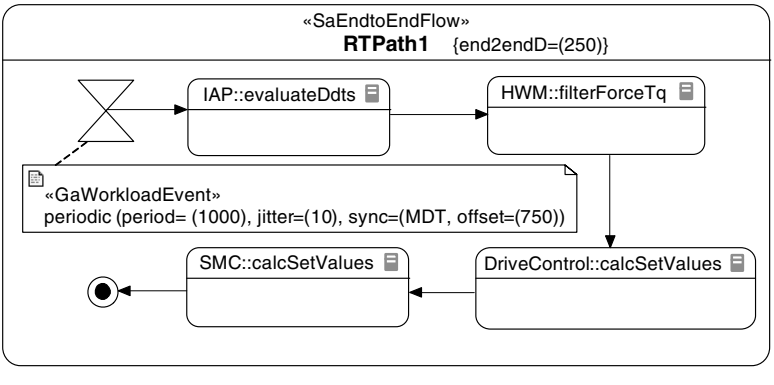


Fig. 8. Real-Time critical path as activity diagram

Telegram. Those telegrams are sent by DSP nodes (see Fig. 1) and contain measured actual values. After the last *DDT* has been received, all *DDTs* are evaluated and stored in shared memory areas by the *IAP* task. This task activates the *HWM* task, which performs a filtering algorithm. The following two tasks execute control algorithms, which generate new nominal values for the drives of the robot and its piezoelectric actuators. All this has to be finished before the deadline, i.e., 250 μs after the *DDT* event (annotated as tagged value *end2endD*). The name of each activitiy in the diagram is composed of the names of the according *SchedulableResource* class and *SaStep* method.

4.3 Model Transformation and Analysis

The UML diagrams including the MARTE annotations can be exported from Rhapsody into an XMI file. XMI is an XML format for metadata interchange, which is standardized by the OMG and is mainly used for the interchange of UML models between different tools. Within the SFB 562, model transformations have been realized that extract relevant information from such XMI file and translate it into representations that allow analyses, e.g., regarding the conformance to real-time requirements. Fig. 9 shows some of the used tools and artifacts in SPEM[4] notation.

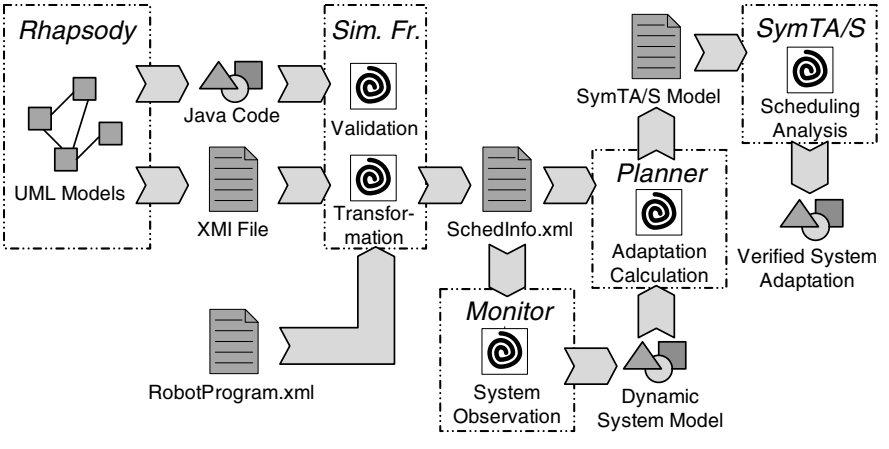


Fig. 9. Model transformations to realize scheduling analysis

The model transformation that transforms the XMI file exported from Rhapsody has been integrated in the simulation framework described in Sec. 2. From within it, an intermediate representation (*SchedInfo.xml*) can be exported, which is used in two different ways, online and offline.

[4] SPEM - Software Process Engineering Metamodel, standardized by the OMG.

4.3.1 Online Approach

The *SchedInfo.xml* file does not only contain information relevant for the scheduling analysis but also quantified data dependencies between *PROSA-X* tasks. Those data dependencies are used for an online optimization and reorganization of the distributed control system, which is based on graph partitioning [24, 12].

Therefore, a monitor that is part of a self-management component (*DistManager* in Fig. 1) uses *SchedInfo.xml* to create the initial instance of a dynamic system model within a QNX shared memory area. This system model is continuously updated by the monitor w.r.t. the topology, tasks within the system, and execution times. Based on an analysis of changes within the dynamic model and data dependencies specified in *SchedInfo.xml*, a planner (also part of *DistManager*) calculates a system adaptation containing a mapping of control tasks to available CPU resources. This candidate for a system adaptation is an architecture variant whose conformance to real-time requirements has to be checked before it is realized. To this end, a scheduling analysis within SymTA/S has been realized, that is performed during the runtime of the system. The planner uses the dynamic system model to generate an according SymTA/S model, which is then analyzed by a SymTA/S server. More details about the self-managed control system utilizing the online analysis can be found in [25].

4.3.2 Offline Approach

The *SchedInfo.xml* as described in the previous subsection contains all tasks of PROSA-X with their timing relevant properties. However, not all combinations of control modules are reasonable. Only a subset of them is used at a time, depending on the current robot task. Such robot task is specified as a skill primitive in an according skill primitive net and specifies concrete layers of control (handled by according control tasks) for each degree of freedom [TMH+05].

The simulation framework is capable to read a skill primitive net (e.g., *RobotProgram.xml* in Fig. 9) in addition to the XMI file exported from Rhapsody and to create separate *SchedInfo.xml* files for the different workload situations of the control system. The planner component is utilized in an offline mode to translate these files into SymTA/S models, to trigger the analysis, and to store the calculated resource utilization and path latencies. An automatically generated report summarizes the conformance or non-conformance of a skill primitive net to the real-time requirements. The files positively analyzed in SymTA/S become part of a data base for the online use of the planner. This way, no online scheduling analysis is necessary when the dynamic system model described in Sec. 4.3.1 represents a workload situation already verified offline.

4.3.3 Analysis in SymTA/S

During its analysis, SymTA/S evaluates if the specified system is schedulable and calculates the resource utilization of processors and buses. First, it determines the worst case and best case response times of all tasks for each resource separately.

Then, those response times together with an *Input Event Model* are used to calculate an *Output Event Model*, which is propagated to dependent resources via event streams [23]. This way a scheduling analysis for distributed systems is realized. Tasks connected via event streams form possible execution paths for which SymTA/S determines the conformance to specified deadlines. If cyclic dependencies exist in the model, it is analyzed iteratively until the results converge.

SymTA/S supports the analysis of worst case response times on a system level but does not contain analysis algorithms for the bus system used in the SFB 562 (IEEE1394, FireWire). However, it can be extended via plugins. A FireWire plugin for SymTA/S has been developed, which delivers best and worst case response times for communication tasks assigned to FireWire bus resources [13]. The plugin has been designed according to the FireWire specification and contains algorithms to analyze asynchronous and isochronous communication.

Although the approach described here has not been used in an early design phase of *PROSA-X*, it is suited for such early scheduling analyses to validate the conformance of a system architecture to real-time requirements. A prerequisite, however, is the availability of architectural details such as best/worst case execution times. The more information about the architecture is available, the better are the results of the scheduling analysis.

5 Conclusion

This article describes several quality assurance measures that have been applied in the development process of a software architecture for parallel kinematic machines. Complexity combined with rigid real-time and safety requirements motivated a model-based engineering approach. Validation, verification, and scheduling analysis approaches have been introduced, which use UML models of structure and behavior of *PROSA-X* to check the conformance to different functional and non-functional requirements.

A simulation framework, which provides functionality of the middleware and the used real-time operating system QNX, was used to validate design decisions and the interoperability of software components at an early development stage. This way, behavioral models could be validated and have served as an executable specification.

A transformation into input for the model checker UPPAAL allowed to verify liveness properties, deadlines, and the correct order of event sequences for critical system components (middleware).

Another transformation was realized to analyze the conformance to real-time requirements on a system level. The scheduling analysis tool SymTA/S has been applied to calculate worst case response times for the periodic real-time critical execution paths within *PROSA-X*. This approach also takes into account the used bus system (FireWire) and is capable to determine the utilization of all resources in the system. This scheduling analysis has also been integrated in *PROSA-X* and is applied during runtime to guarantee a safe reconfiguration of the control system under changing external conditions (robot program, topology).

Further quality assurance measures have been applied, which have not been mentioned here [26]. Skill primitive nets containing robot programs have been transformed to be formally analyzed within the model checker HyTech [27]. In a model-based testing approach, the UML models of *PROSA-X* have served as a basis to determine time optimal test cases for several coverage criteria, using the model checker UPPAAL [28, 29, 30].

Acknowledgements. The authors would like to thank the DFG (German Research Foundation) for the financial support of the collaborative research center 562 "Robotic Systems for Handling and Assembly", and QNX Software Systems as well as the Symtavision GmbH for the grant of free software licenses.

References

1. Merlet, J.-P.: Parallel Robots. Kluwer Academic Publishers, Dordrecht (2000)
2. Kohn, N., Kolbus, M., Reisinger, T., Diethers, K., Steiner, J., Thomas, U.: Prosa - a generic control architecture for parallel robots. In: Proceedings of Mechatronics & Robotics, pp. 55–61. Sascha Eysoldt Verlag, Aachen (2004)
3. Finkemeyer, B.: Robotersteuerungsarchitektur auf der Basis von Aktionsprimitiven. Ph.D. thesis, Technische Universität Carolo-Wilhelmina zu Braunschweig (2004)
4. Diethers, K., Finkemeyer, B., Kohn, N.: Realizing open control software for high dynamic processes with a middleware. it - Information Technology 46(1), 39–47 (2004)
5. Finkemeyer, B., Kröger, T., Wahl, F.M.: A Middleware for High-Speed Distributed Real-Time Robotic Applications. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 193–212. Springer, Heidelberg (2010)
6. Kohn, N., Varchmin, J.-U., Steiner, J., Goltz, U.: Universal communication architecture for high-dynamic robot systems using qnx. In: Proceedings of International Conference on Control, Automation, Robotics and Vision (ICARCV 8th), vol. 1, pp. 205–210. IEEE Computer Society, Kunming (2004)
7. Dadji, Y., Michalik, H., Kohn, N., Steiner, J., Beckmann, G., Möglich, T., Varchmin, J.U.: A Communication Architecture for Distributed Real-Time Robot Control. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 213–231. Springer, Heidelberg (2010)
8. Thomas, U., Wahl, F.M.: Sensor guided execution of robot tasks based on skill primitives. In: Robotic 2002, pp. 71–77 (2002)
9. Thomas, U., Maaß, J., Hesselbach, J., Wahl, F.M.: Towards a new concept of robot programming in high speed assembly applications. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2005, pp. 3932–3938 (2005)
10. Kolbus, M., Reisinger, T., Maaß, J.: Robot Control based on Skill Primitives. In: Robotics and Applications: Sixth IASTED International Conference Proceedings (2005)
11. Maaß, J., Kohn, N., Hesselbach, J.: Open modular robot control architecture for assembly using the task frame formalism. International Journal of Advanced Robotic Systems 3(1), 1–10 (2006)
12. Maaß, J., Steiner, J., Amado, A., Hesselbach, J., Huhn, M., Raatz, A.: Self-management in a control architecture for parallel kinematic robots. In: Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2008, Brooklyn, New York, USA (2008)
13. Steiner, J., Hagner, M., Goltz, U.: Runtime analysis and adaptation of a hard real-time robotic control system. Journal of Computers (JCP) 2(10), 18–27 (2007)

14. Steiner, J., Huhn, M., Mücke, T.: Model based quality assurance and self-management within a software architecture for parallel kinematic machines. In: Proceedings of the IEEE 3rd International Conference on Mechatronics (ICM 2006), pp. 55–60. IEEE Computer Society, Budapest (2006)
15. Diethers, K., Goltz, U., Vocke, S.: Analysis of real-time systems modeled by uml-statecharts. In: Proceedings of the First International Colloquium of SFB 562, Robotic Systems for Handling and Assembly (2002)
16. DeMarco, T.: Structured analysis and system specification. Prentice-Hall software series. Yourdon Press, New York (1978)
17. Diethers, K., Goltz, U., Huhn, M.: Model checking uml statecharts with time. In: Workshop on Critical Systems Development with UML (CSDUML 2002), pp. 35–52 (2002)
18. Diethers, K., Huhn, M.: Vooduu: verification of object-oriented designs using uppaal. In: Jensen, K., Podelski, A. (eds.) TACAS 2004. LNCS, vol. 2988, pp. 139–143. Springer, Heidelberg (2004)
19. Diethers, K.: Werkzeuggestützte formale Analyse von Echtzeitsystemen. Ph.D. thesis, Technical University of Braunschweig, Germany (2006)
20. Larsen, K.G., Pettersson, P., Yi, W.: Uppaal in a nutshell. International Journal on Software Tools for Technology Transfer 1(1+2), 134–152 (1997)
21. Object Management Group. Uml profile for modeling and analysis of real-time and embedded systems (marte), beta 2, omg adopted specification ptc/08-06-09 (2009), http://www.omgmarte.org/Documents/Specifications/08-06-09.pdf (Cited December 16, 2009)
22. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science 126(2), 183–235 (1994)
23. Henia, R., Hamann, A., Jersak, M., Racu, R., Richter, K., Ernst, R.: System level performance analysis - the symta/s approach. IEEE Proceedings Computers and Digital Techniques 152(2), 148–166 (2005)
24. Steiner, J., Amado, A., Goltz, U., Hagner, M., Huhn, M.: Engineering self-management into a robot control system. In: Proceedings of 3rd International Colloquium of the Collaborative Research Center 562, pp. 113–125 (2008)
25. Steiner, J., Goltz, U., Maaß, J.: Self-Management within a Software Architecture for Parallel Kinematic Machines. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 355–371. Springer, Heidelberg (2010)
26. Steiner, J., Diethers, K., Mücke, T., Goltz, U., Huhn, M.: Rigorous tool-supported software development of a robot control system. In: Last, P., Budde, C., Wahl, F.M. (eds.) Proceedings of 2nd International Colloquium of the Collaborative Research Center 562. Fortschritte in der Robotik, vol. 9, pp. 137–151. Shaker Verlag, Aachen (2005)
27. Diethers, K., Firley, T., Kroeger, T., Thomas, U.: A new framework for task oriented sensor based robot programming and verification. In: Proceedings of IEEE International Conference on Advanced Robotics, Coimbra, Portugal (2003)
28. Mücke, T., Huhn, M.: Generation of optimized testsuites for uml statecharts with time. In: Groz, R., Hierons, R.M. (eds.) TestCom 2004. LNCS, vol. 2978, pp. 128–143. Springer, Heidelberg (2004)
29. Mücke, T., Huhn, M.: Minimizing test execution time during test generation. In: IFIP Working Conference on Software Engineering Techniques (SET 2006), Springer, Heidelberg (2006)
30. Huhn, M., Mücke, T.: Comparing heuristics for model based testsuite generation. In: Model-Based Development of Embedded Systems (MBEES 2006) (2006)

Part IV
Adaptronics and Components

Modelling of Piezoceramic Patches for Augmenting Modal Structural Models with Flat Actuator Devices

Michael Rose

Abstract. This section discusses the modelling of piezoceramic patches, which are used as structural actuators in highly accelerated light weight robot structures to achieve short cycle times in assembly tasks. Such devices in combination with suitable control algorithms can increase the structural damping significantly. The subcomponents of parallel structures, augmented with such actuator devices, can be modelled with the Finite Element method. After a modal decomposition, the mass and stiffness coefficients can be adjusted by a special so-called modal correction method for the applied flat piezoceramic devices in combination with their electromechanical coupling. The underlying electromechanical triangle element is also discussed.

1 Introduction

Parallel robot systems are usually constructed with the heavy weights of the drives being concentrated at the immovable base platform. The moving chains thus are significantly reduced in weight regarding serial robots. Though the workspace is often quite limited, the big advantage is a potential higher acceleration with reduced cycle times in handling and assembly tasks, leading to very fast and responsive machines. This effect is often intensified through lightweight materials like carbon fibre composites (*CFC*), which are used to build up rods, cranks, and other parts of the movable mechanism. The high stiffness of these materials in combination with the parallel setup induces a very stiff manipulator behaviour in large areas of the workspace, but unfortunately CFC features poor damping. This induces unwanted vibrations, and these unpleasent oscillations of the end-effector can have a major impact on the precision of such robot systems.

Michael Rose
German Aerospace Center (DLR), Institute of Composite Structures and Adaptive Systems, Lilienthalplatz 7, 38108 Braunschweig, Germany
e-mail: michael.rose@dlr.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 393–411.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

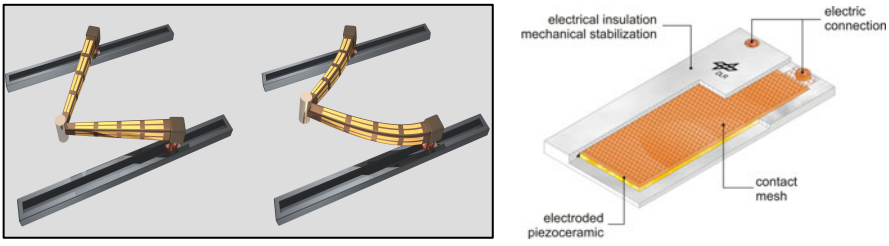


Fig. 1. a Simple planar elastic robot structure of type PARAPLACER in undeformed/deformed state with attached patch actuators. **b** Layered assembly of a standard patch actuator.

One approach to increase the structural damping is to use smart materials as actuators in combination with suitable control laws. This adaptronic solution with active elastic components has been studied in the SFB 562 for several years now. It is desirable to have a system model including the rigid dynamics and the structural dynamics due to the elasticities of the individual chain components, either for simulation purposes or to generate model-based control algorithms for vibration suppression. In Fig. 1, the basic idea is visualized for a simple planar robot system of so-called PARAPLACER-type. The idea is to use flat piezoelectric devices as shown in the figure. They are glued or embedded into suitable components of the moving structure and introduce forces and moments if an electric potential is applied to the electrodes. Their coupling into the structure is significantly different from that of customary stack actuators (Fig. 2). Most often such elastic components with active devices are modelled as complete finite element models. This has the drawback that optimal placements of the active devices can only be calculated by large changements in these discrete models to reposition the devices in each iteration step. The new approach, which will be discussed in this article, enables a rapid placement of the active devices on an elastic component by the so-called modal coupling method, discussed in Sect. 5. To use this method, the electromechanical coupling of the piezoceramic patch actuators must be acurately described. This will also be discussed here in detail.

2 Related Work

The approach discussed in this article is heavily based upon the finite element method. There is a huge amount of literature covering this topic. A general treatment of this method can be found in many textbooks like [1] to cite a milestone work on this area. Special treatment of the piezoelectric coupling within shell elements is extensively discussed in [2] for curved shell elements, oriented in principal curvature directions (e.g. parts of spheres, cones, cylinders, planes). In most commercial available finite element programs like ANSYS©, piezoelectric coupling is available. In ANSYS©, a sophisticated shell element was implemented, allowing several layers of composites to consist of active material (see [3]). Other approaches as in [4] use so-called mixed formulations in the variational sense to implement

accurate coupling elements. In contrast to these notable results, the ultimate goal in the modal coupling approach is not precision in estimating strains and stresses, but to have an acceptable element with respect to the displacement accuracy, which can be rapidly added to exported models of elastic components in foreign environments. Simplicity in the application is a major goal. Therefore, a new triangle element with piezoelectric coupling was developed and applied in the matrix-based software MATLAB© to modal decomposed substructures. The details of this element are extensively discussed in [5]. One important aspect on using modal decompositions to reduce the degrees of freedom is the choice of modes if such elastic components are used in multibody simulations. A detailed analysis of this non-intuitive topic with focus on fatigue analysis is given in [6]. To increase the accuracy of this inherently approximative approach, even static deformations and dynamic deformation modes at non-resonant frequencies should be added in some cases with carefully choosen boundary conditions.

General treatment of adaptive structures regarding modelling and control can be found in established text books like [7, 8]. The structural control of the TRIGLIDE-demonstrator (Fig. 3b) by a mode switching control algorithm, based on system identification at discrete locations in the workspace, is discussed in [9].

3 Vibrations of Lightweight Parallel Robot Structures

Lightweight mechanisms perform elastic deformations, which are superimposed on the large non-linear rigid body translations and rotations. Each rigid component of such mechanisms can be observed in a coordinate reference system $\mathcal{O}_B$, attached to some nodes of this mechanical substructure. If the elastic deformations are described by a linear superposition of some modes from a suitable modal analysis and additional degrees of freedom (*DOF*) are choosen for the rigid body movements, the dynamic equations are fully coupled. For sufficiently stiff components, the structural deformations are of lower order than the rigid motions. Therefore, it is possible to get satisfactory results if the multibody system is solved with rigid body subcomponents and the boundary forces, as well as the inertia gyroscopic and centrifugal terms, are used as disturbances in the structural elastics part of the equations. All vibrations, which are calculated in the following discussion, have to be treated as overlays to the already calculated rigid body movements of the parallel robot system.

3.1 Elastic Equations of Mechanical Structures

If $\boldsymbol{u}$ denotes the elastic deformation of the component with respect to $\mathcal{O}_B$, the force bilance of the linearized structural dynamic equations is given by

$$\hat{\boldsymbol{M}}\,\ddot{\boldsymbol{u}} + \hat{\boldsymbol{D}}\,\dot{\boldsymbol{u}} + \hat{\boldsymbol{K}}\,\boldsymbol{u} = \hat{\boldsymbol{f}}. \tag{1}$$

If matrices or vectors are based on physical DOF's, a hat accent is used to distinguish them from their modal representation. The latter ones are written without accentuation. In $\hat{\boldsymbol{f}}$ all gyroscopic, centrifugal and boundary force terms are collected. The mass matrix $\hat{\boldsymbol{M}}$ and stiffness matrix $\hat{\boldsymbol{K}}$ can be obtained either analytically or through a finite element program. Usually, the damping matrix is approximated by some modal damping coefficients due to the rough modelling precision of material damping in real structures. In Fig. 3, a very detailed finite element model of an elastic crank is displayed. To incorporate some active devices into such a substructure, a mesh must be exported from the finite element program, which describes suitable surface parts for mounting such devices. The elastic DOF's are nodal translations and rotations of mesh nodes and other boundary nodes connecting the body to neighbouring bodies. The boundary nodes need not lay on the mesh if a suitable reduction of DOF's is performed in the finite element program.

3.2 Piezoceramic Patches as Structural Actuators

If the mass and stiffness effects of the stack and patch actuators are neglected, their actuation influence can be described by suitable force pairs or line moments, as shown in Fig. 2. Therefore, with stack actuators, a discrete external force pair can be added to $\hat{\boldsymbol{f}}$ of eq. (1). This force pair is proportional to the applied electric voltage if non-linear effects in the constitutive equations are ignored. The integral effect of the active material strains is given by a positive or negative elongation with respect to the two boundary end nodes.

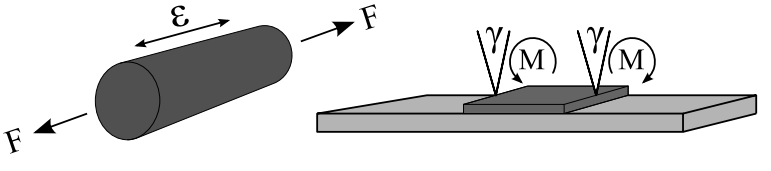


Fig. 2. a Equivalent force pair for stack actuator. **b** Equivalent line moments for patch actuator.

In a similar way the stresses of a flat piezo patch can be equivalently generated by virtual line bending moments at the boundary of the area, covered by such devices. Another elegant way to describe these stresses in classical finite element programs uses the temperature as voltage value. The temperature strain coupling coefficients α are interpreted as the piezoelectric coupling coefficients in this context. Of course, only the actuation properties due to the inverse piezoelectric effect are modeled by this trick. The sensor properties — a result of the direct piezoelectric effect — are neglected. To account for the fully coupled equations, the implemented piezoelectric DOF's of a modern finite element program have to be used for accurate models.

3.3 Substructure Decomposition

Figure 3 displays a very rough finite element model of the TRIGLIDE-demonstrator, which is discussed in other chapters of this book thoroughly. The fully assembled model is non-linearly dependent on the actual workspace location of the manipulator. This complicates the treatment of elastic vibrations, because there is no possibility to solve a constant linear system model in the frequency domain to obtain a suitable control law for vibration suppression. In fact the matrix coefficients of eq. (1) for the fully assembled parallel robot system are time dependent values and must be recalculated for each individual trajectory. One solution to obtain a state space vibration control is presented in this book. It uses an interpolation strategy for individual state space control matrices, obtained at different grid points of the work space. The plant matrices $\hat{\boldsymbol{M}}$, $\hat{\boldsymbol{D}}$ and $\hat{\boldsymbol{K}}$ are indirectly estimated by a system identification of the real demonstrator. For simulation purposes it is possible to assemble these matrices from the elastic subcomponents according to the positions and orientations of them in space at each time t. This allows a simulation of the controlled system and enables the estimation of the achievable vibration suppression a priori, before the real structure has been fully designed and built.

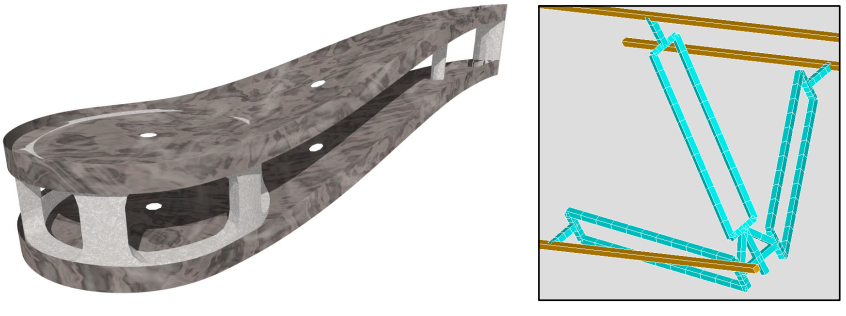


Fig. 3. a Detailed finite element model of a crank substructure. **b** Simple finite element assembly of the TRIGLIDE-robot.

4 Flat Piezoceramic Patch Devices

The development of bending triangles as finite elements with respect to the Kirchhoff plate bending theory is extensively discussed in the literature. A very elaborate analysis of the advantages and disadvantages of different approaches can be found in [10]. Triangular elements can be problematic and suffer from severe performance penalties if naively constructed. For vibration analysis, it is the right element type though, because stresses are not of main concern and the simple application to triangular meshes is a big advantage in placement strategies for the piezo modules. As a result in [10], so-called compatible elements are less efficient then suitable designed incompatible ones, but they are guaranteed to converge. Therefore, in [5] detailed derivations of two different triangular elements are presented. The first one fulfills

the compatibility criterion and is based on the classical Clough-Tocher macro element. The second one is an incompatible element, based on [11], which is called Specht's element for brevity here. In practice the latter has been approved and will be explained in the following, because it is also somewhat simpler in derivation as the former element type.

4.1 Full Equations for Thin Bending Structures

The compatibility criterion with respect to the Kirchhoff plate theory is characterized by C^1-continuity of the transversal displacement w. Combined with the membrane displacements u and v, the kinematic ansatz is given by

$$\boldsymbol{u} = \begin{bmatrix} u - z\,w_{,x} \\ v - z\,w_{,y} \\ w \end{bmatrix}, \quad z_m - \frac{h}{2} \le z \le z_m + \frac{h}{2}.$$

The displacements $u(x,y)$, $v(x,y)$ and $w(x,y)$ and the rotations of the cross sections $\Phi_x(x,y) = w_{,y}(x,y)$ and $\Phi_y(x,y) = -w_{,x}(x,y)$ are related to the neutral plane at $z = 0$. The center of the piezo modul of thickness h is located at the signed distance z_m from the neutral plane. This leads to the linearized strain terms

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} u_{,x} - z\,w_{,xx} \\ v_{,y} - z\,w_{,yy} \\ u_{,y} + v_{,x} - 2z\,w_{,xy} \end{bmatrix}. \tag{2}$$

The plain stress conditions $\sigma_z = \tau_{xz} = \tau_{yz} = 0$ substitute the inconsistent kinematic restrictions $\varepsilon_z = \gamma_{xz} = \gamma_{yz} = 0$ due to the limitations of ansatz (2). The choosen law for the electrically active material, reduced to the relevant stress and strain components, is given by

$$\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \frac{1}{E} \begin{bmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 2(1+\nu) \end{bmatrix} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} + \begin{bmatrix} d_{31} \\ d_{31} \\ 0 \end{bmatrix} E_3, \tag{3}$$

whereas the electric field is assumed to possess only one component E_3 in z-direction, and the passive material properties should be isotropic. This can be transformed into

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \underbrace{\frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}}_{=\boldsymbol{C}} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} - \underbrace{\begin{bmatrix} e^{\star}_{31} \\ e^{\star}_{31} \\ 0 \end{bmatrix}}_{=\boldsymbol{E}} E_3 \quad \text{with} \quad e^{\star}_{31} = \frac{E}{1-\nu} d_{31}. \tag{4}$$

The star in $e^{\star}_{31}$ indicates that the coefficient does not correspond directly to the three dimensional material tensor e_{ij}; the plain stress state is already accounted for in that coefficient. The matrix $\boldsymbol{C}$ can also be represented by engineer constants as

$$\boldsymbol{C} = \frac{E}{2(1-\nu^2)} \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1+\nu & 0 & 0 \\ 0 & 1-\nu & 0 \\ 0 & 0 & 1-\nu \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Other piezo modules based on fibre ceramics can be modelled with a different set of material matrices $\boldsymbol{C}$ and $\boldsymbol{E}$, as given by the more general relations

$$\boldsymbol{C} = E^\star \begin{bmatrix} 1 & \nu & 0 \\ \nu & \nu_E & 0 \\ 0 & 0 & \frac{1-\nu_G\,\nu}{2} \end{bmatrix} \quad \text{and} \quad \boldsymbol{E} = e^\star \begin{bmatrix} 1 \\ \nu_e \\ 0 \end{bmatrix} = \begin{bmatrix} e_x^\star \\ e_y^\star \\ 0 \end{bmatrix} \tag{5}$$

with three additional dimensionless parameters ν_E, ν_G, and ν_e being one in the special isotropic case. $\boldsymbol{C}$ and $\boldsymbol{E}$ as in eq. (5) are based on the fibre orientation $\varphi = 0$. To obtain representations for arbitrary fibre angles φ these equations have to be rotated according to the standard tensor laws. If rotational inertia at infinitesimal level is neglected, the mass matrix $\boldsymbol{M_p}$, stiffness matrix $\boldsymbol{K_p}$, as well as the coupling vector $\boldsymbol{f_p}$ of the ceramic module can be calculated from the variational formulation

$$\boldsymbol{\delta q}^T \left(\boldsymbol{M_p}\,\ddot{\boldsymbol{q}} + \boldsymbol{K_p}\,\boldsymbol{q} - \boldsymbol{f_p} \underbrace{h\,E_3}_{=U} \right) = \int_V \left(\rho \begin{bmatrix} \delta u \\ \delta v \\ \delta w \end{bmatrix}^T \begin{bmatrix} \ddot{u} \\ \ddot{v} \\ \ddot{w} \end{bmatrix} + \begin{bmatrix} \delta\varepsilon_x \\ \delta\varepsilon_y \\ \delta\gamma_{xy} \end{bmatrix}^T \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} \right) dV. \tag{6}$$

Application of the Gauss divergence theorem enables some simplifications regarding the coupling vector $\boldsymbol{f_p}$. With the outer normal vector $\boldsymbol{n}$ and the active area A

$$\begin{aligned} \boldsymbol{\delta q}^T \boldsymbol{f_p} &= \frac{e_{31}^\star}{h} \int_V (\delta\varepsilon_x + \delta\varepsilon_y)\, dV = \frac{e_{31}^\star}{h} \int_{z_m-h/2}^{z_m+h/2} \int_{\partial A} \begin{bmatrix} \delta u - z\,\delta w_{,x} \\ \delta v - z\,\delta w_{,y} \end{bmatrix}^T \boldsymbol{n}\, ds\, dz \\ &= e_{31}^\star \int_{\partial A} \left(\begin{bmatrix} \delta u \\ \delta v \end{bmatrix}^T \boldsymbol{n} - z_m \frac{\partial \delta w}{\partial \boldsymbol{n}} \right) ds. \end{aligned} \tag{7}$$

Accordingly, the anisotropic constitutive material law (5) leads to

$$\boldsymbol{\delta q}^T \boldsymbol{f_p} = \frac{1}{h} \int_V (e_x^\star \delta\varepsilon_x + e_y^\star \delta\varepsilon_y)\, dV = \int_{\partial A} \begin{bmatrix} e_x^\star\, n_x \\ e_y^\star\, n_y \end{bmatrix}^T \left(\begin{bmatrix} \delta u \\ \delta v \end{bmatrix} - z_m \begin{bmatrix} \delta w_{,x} \\ \delta w_{,y} \end{bmatrix} \right) ds.$$

To simplify the discussion, anisotrop material laws will not be discussed further. For the standard modules, which are based on the d_{31}-effect, they are not needed. A generalization to the anisotropic case of all derivations is straight forward. Only one subtle point needs careful consideration. The fibre angle of anisotropic elements has to be specified in each finite element of the structural mesh. This book keeping problem can be cumbersome if arbitrary meshes are handled. In addition to the material stiffness matrix $\boldsymbol{C}$ from eq. (5), the coupling vector $\boldsymbol{f_p}$ has to be modified in the anisotropic case as well.

4.2 Barycentric Coordinates for Triangular Elements

In Fig. 4, the barycentric coordinates (r, s, t) are defined with respect to the triangle $P_3P_1P_2$. Componentwise, the two dimensional point vectors

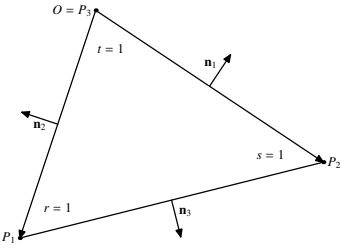


Fig. 4. Barycentric coordinates in triangle $P_3P_1P_2$.

$$\boldsymbol{p} = \begin{bmatrix} x \\ y \end{bmatrix} = r\,\boldsymbol{p}_1 + s\,\boldsymbol{p}_2, \quad \boldsymbol{p}_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \quad \boldsymbol{p}_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, \quad 0 \le r, s, t$$

with $r + s + t = 1$ are introduced. In matrix notation this results in

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix}, \quad \begin{bmatrix} r \\ s \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} y_2 & -x_2 \\ -y_1 & x_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad \text{with} \quad \Delta = x_1y_2 - x_2y_1.$$

The outer normals at each side are given by

$$\boldsymbol{n}_1 = \begin{bmatrix} -y_2 \\ x_2 \end{bmatrix}, \ \boldsymbol{n}_2 = \begin{bmatrix} y_1 \\ -x_1 \end{bmatrix}, \ \boldsymbol{n}_3 = \begin{bmatrix} y_2 - y_1 \\ x_1 - x_2 \end{bmatrix}.$$

A function f, defined on the area of the triangle, is a function of (x, y) or (r, s) due to the barycentric condition $t = 1 - r - s$. The abbreviations

$$f_{,x} = \frac{\partial f}{\partial x}, \ f_{,y} = \frac{\partial f}{\partial y}, \ f_{,r} = \frac{\partial f}{\partial r} - \frac{\partial f}{\partial t}, \ f_{,s} = \frac{\partial f}{\partial s} - \frac{\partial f}{\partial t}$$

will be used as well as similar ones for the second derivatives. They can be transformed by

$$\begin{bmatrix} f_{,r} \\ f_{,s} \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} \begin{bmatrix} f_{,x} \\ f_{,y} \end{bmatrix}, \quad \begin{bmatrix} f_{,rr} & f_{,rs} \\ f_{,rs} & f_{,ss} \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} \begin{bmatrix} f_{,xx} & f_{,xy} \\ f_{,xy} & f_{,yy} \end{bmatrix} \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix}$$

or inversely by

$$\begin{bmatrix} f_{,x} \\ f_{,y} \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} y_2 & -y_1 \\ -x_2 & x_1 \end{bmatrix} \begin{bmatrix} f_{,r} \\ f_{,s} \end{bmatrix}, \quad \begin{bmatrix} f_{,xx} \\ f_{,yy} \\ 2f_{,xy} \end{bmatrix} = \frac{1}{\Delta^2} \begin{bmatrix} y_2^2 & y_1^2 & -y_1y_2 \\ x_2^2 & x_1^2 & -x_1x_2 \\ -2x_2y_2 & -2x_1y_1 & x_1y_2 + x_2y_1 \end{bmatrix} \begin{bmatrix} f_{,rr} \\ f_{,ss} \\ 2f_{,rs} \end{bmatrix}.$$

Therefore, the transformation of $(u_{,r}, u_{,s})^T$, $(v_{,r}, v_{,s})^T$ and $(w_{,r}, w_{,s})^T$ into global strain values with respect to (x, y) is given by the matrix factors

$$\boldsymbol{D}_{12}^{u} = \frac{1}{\Delta}\begin{bmatrix} y_2 & -y_1 \\ 0 & 0 \\ -x_2 & x_1 \end{bmatrix},\ \boldsymbol{D}_{12}^{v} = \frac{1}{\Delta}\begin{bmatrix} 0 & 0 \\ -x_2 & x_1 \\ y_2 & -y_1 \end{bmatrix},$$

$$\boldsymbol{D}_{12}^{w} = \frac{1}{\Delta^2}\begin{bmatrix} y_2^2 & y_1^2 & -y_1y_2 \\ x_2^2 & x_1^2 & -x_1x_2 \\ -2x_2y_2 & -2x_1y_1 & x_1y_2 + x_2y_1 \end{bmatrix}. \quad (8)$$

In addition, the derivative with respect to the outer normal $\boldsymbol{n}$ can be written as

$$\frac{\partial f}{\partial \boldsymbol{n}} = \frac{1}{\|\boldsymbol{n}\|}\begin{bmatrix} n_x \\ n_y \end{bmatrix}^T \begin{bmatrix} f_{,x} \\ f_{,y} \end{bmatrix} = \frac{1}{\Delta\|\boldsymbol{n}\|}\begin{bmatrix} n_x \\ n_y \end{bmatrix}^T \begin{bmatrix} y_2 & -y_1 \\ -x_2 & x_1 \end{bmatrix}\begin{bmatrix} f_{,r} \\ f_{,s} \end{bmatrix}. \quad (9)$$

Using the identities

$$\begin{bmatrix} t_y \\ -t_x \end{bmatrix}^T \begin{bmatrix} y_2 & -y_1 \\ -x_2 & x_1 \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}^T \begin{bmatrix} x_2 & -x_1 \\ y_2 & -y_1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} t_x \\ t_y \end{bmatrix}^T \begin{bmatrix} x_2 - x_1 & x_2 + x_1 \\ y_2 - y_1 & y_2 + y_1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

which are valid for arbitrary values t_x, t_y, and the abbreviations

$$\sigma_1 = \frac{\boldsymbol{p}_1^T\boldsymbol{p}_2}{\|\boldsymbol{p}_2\|^2},\ \sigma_2 = \frac{\boldsymbol{p}_1^T\boldsymbol{p}_2}{\|\boldsymbol{p}_1\|^2},\ \sigma_3 = \frac{\|\boldsymbol{p}_2\|^2 - \|\boldsymbol{p}_1\|^2}{\|\boldsymbol{p}_2 - \boldsymbol{p}_1\|^2}, \quad (10)$$

the normal derivatives at each side can be expressed as

$$\frac{\partial f}{\partial \boldsymbol{n}_1} = \frac{\|\boldsymbol{p}_2\|}{\Delta}\begin{bmatrix} -1 \\ \sigma_1 \end{bmatrix}^T \begin{bmatrix} f_{,r} \\ f_{,s} \end{bmatrix},\quad \frac{\partial f}{\partial \boldsymbol{n}_2} = \frac{\|\boldsymbol{p}_1\|}{\Delta}\begin{bmatrix} \sigma_2 \\ -1 \end{bmatrix}^T \begin{bmatrix} f_{,r} \\ f_{,s} \end{bmatrix},$$

$$\frac{\partial f}{\partial \boldsymbol{n}_3} = \frac{\|\boldsymbol{p}_2 - \boldsymbol{p}_1\|}{2\Delta}\begin{bmatrix} 1 \\ \sigma_3 \end{bmatrix}^T \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} f_{,r} \\ f_{,s} \end{bmatrix}. \quad (11)$$

4.3 Evaluation of Integrals in Barycentric Coordinates

All integrals with respect to triangle areas and corresponding barycentric coordinates can be evaluated analytically for polynomials. This is possible using the monomial integration rule

$$\int_{OP_1P_2} r^a s^b t^c \, dA = \Delta \cdot \Psi(r^a s^b t^c) \quad \text{with}$$

$$\Psi(r^a s^b t^c) = \frac{a!b!c!}{(a+b+c+2)!},\ a, b, c \in \mathbb{N}_0, \quad (12)$$

which can be derived by induction and partial integration. Although all integrands for the finite element are of polynomial form, exact integration will not be used for the stiffness terms. Reduced integration by suitable quadrature formulas often increases the quality of obtained solutions, and this turns out to be the case for Specht's element as well. The Gauss quadrature formula for integration in z-direction

$$\int_{z_m-h/2}^{z_m+h/2} f(z)\,dz \approx \frac{h}{2}\left(f(z_m - \frac{h}{2\sqrt{3}}) + f(z_m + \frac{h}{2\sqrt{3}})\right) \tag{13}$$

gives exact results for cubic polynomials with respect to z.

4.4 Membrane Displacements

To avoid finite element nodes at the midpoints of the triangle sides, membrane displacements are only interpolated linearly as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} r & s & t \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma}_u & \boldsymbol{\gamma}_v \end{bmatrix} = \begin{bmatrix} r & s & t \end{bmatrix} \begin{bmatrix} \gamma_{u1} & \gamma_{v1} \\ \gamma_{u2} & \gamma_{v2} \\ \gamma_{u3} & \gamma_{v3} \end{bmatrix}.$$

A quadratic interpolation order is also easily implementable, but is avoided here to keep mesh processing as simple as possible. The inertia terms with respect to membrane displacements are now obtained as

$$h\rho \int_{OP_1P_2} \delta u \cdot u\,dA = h\rho \frac{\Delta}{24} \delta\boldsymbol{\gamma}_u{}^T \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \boldsymbol{\gamma}_u,$$

with the piezo patch density ρ and its thickness h. A similar result holds in y-direction by substitution of u with v.

4.5 Specht's Incompatible Triangular Element

Zienkiewicz [1] and Argyris [10] have discussed incompatible triangular elements for plate bending at length. A subtle variation of their elements is discussed in [11], which proved to be very accurate in practice and will be the foundation of the developed piezoelectric finite element, which will be denoted here as Specht's incompatible triangular element. It is based on a cubic set of form functions with zero and first order derivatives at the triangle vertices. There are nine degrees of freedom (f, $f_{,x}$ and $f_{,y}$ at each vertex), but ten monomials for a complete cubic polynom. The linear and quadratic terms $\{r, s, t, rs, st, tr\}$ must all be present to obtain correct bending elements. A critical aspect is the choice of three form functions as replacement of the four cubic monomials $\{r^2s, s^2t, t^2r, rst\}$. This is done in [11] by a condition, which fulfills the so-called patch test, using the three modified functions

$$\begin{bmatrix} r^2 s \\ s^2 t \\ t^2 r \end{bmatrix} + \boldsymbol{K} \begin{bmatrix} r^2 st \\ rs^2 t \\ rst^2 \end{bmatrix}$$

with additional terms of fourth order as elementary components of the form functions. The latter ones are given by

$$\boldsymbol{g} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \\ g_7 \\ g_8 \\ g_9 \end{bmatrix} = \underbrace{\begin{bmatrix} r(1+t-s) \\ -r\,t \\ 0 \\ s(1+r-t) \\ r\,s \\ -r\,s \\ t(1+s-r) \\ 0 \\ s\,t \end{bmatrix}}_{=\boldsymbol{g}^k} + \underbrace{\begin{bmatrix} 2 & 0 & -2 \\ -1 & 0 & 1 \\ 1 & 0 & 0 \\ -2 & 2 & 0 \\ -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & -2 & 2 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}}_{=\boldsymbol{V}^T} \boldsymbol{g}^v \quad \text{with} \tag{14}$$

$$\boldsymbol{g}^v = \begin{bmatrix} r^2 s \\ s^2 t \\ t^2 r \end{bmatrix} + \frac{r\,s\,t}{2} \underbrace{\begin{bmatrix} 3(1+\mu_3) & 3\mu_3 - 1 & 1 - 3\mu_3 \\ 1 - 3\mu_1 & 3(1+\mu_1) & 3\mu_1 - 1 \\ 3\mu_2 - 1 & 1 - 3\mu_2 & 3(1+\mu_2) \end{bmatrix}}_{=2\boldsymbol{K}} \begin{bmatrix} r \\ s \\ t \end{bmatrix}.$$

and the scalar values in $\boldsymbol{K}$ are abbreviations for

$$s_1 = \|\boldsymbol{p}_2\|, \qquad s_2 = \|\boldsymbol{p}_1\|, \qquad s_3 = \|\boldsymbol{p}_2 - \boldsymbol{p}_1\|,$$
$$\mu_1 = \frac{s_2^2 - s_3^2}{s_1^2}, \qquad \mu_2 = \frac{s_3^2 - s_1^2}{s_2^2}, \qquad \mu_3 = \frac{s_1^2 - s_2^2}{s_3^2}.$$

The μ_i and σ_i from eq. (10) are dependent from each other by

$$\sigma_1 = \frac{1+\mu_1}{2}, \quad \sigma_2 = \frac{1-\mu_2}{2}, \quad \sigma_3 = \mu_3. \tag{15}$$

It can be verified easily that these form functions comply with the requirements

$$\begin{aligned} g_1 = g_{2,r} = g_{3,s} &= 1 \quad \text{for } r = 1, \\ g_4 = g_{5,r} = g_{6,s} &= 1 \quad \text{for } s = 1, \\ g_7 = g_{8,r} = g_{9,s} &= 1 \quad \text{for } t = 1, \\ g_j = g_{j,r} = g_{j,s} &= 0 \quad \text{else for } t = 1 \text{ or } r = 1 \text{ or } s = 1. \end{aligned} \tag{16}$$

In addition, all components from $\boldsymbol{g}$ are of maximal cubic order at each side of the triangle. Due to the hermite interpolation conditions between neighbouring triangular elements, the combined functions of different elements are continuous and have continuous tangential derivatives at the connecting sides. But the normal derivatives won't be continuous in general, therefore this element will be a so-called incompatible element and not of class C^1. In order to fulfill the patch test, Specht used the weaker condition that the integral

$$\int_{\boldsymbol{p}_a}^{\boldsymbol{p}_b} \frac{\partial f}{\partial \boldsymbol{n}}\, ds$$

at each common side of two elements with vertices $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$ will always be equal (but of opposite sign). Therefore, the pointwise equality of C^1-continuity is weakened by this integral condition. In [5], this is validated for the form functions (14).

In order to convert the global degrees of freedom

$$\boldsymbol{\gamma}_w{}^T = \left[\gamma_1\ \gamma_{1x}\ \gamma_{1y}\ \gamma_2\ \gamma_{2x}\ \gamma_{2y}\ \gamma_3\ \gamma_{3x}\ \gamma_{3y}\right] \tag{17}$$

into suitable coefficients for $\boldsymbol{g}$, the transformation

$$w = \boldsymbol{g}^T \boldsymbol{T}_3^w \boldsymbol{\gamma}_w, \quad \delta w = \boldsymbol{g}^T \boldsymbol{T}_3^w \boldsymbol{\delta\gamma}_w \quad \text{with} \quad \boldsymbol{T}_3^w = \begin{bmatrix} \boldsymbol{T}_{12} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}_{12} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{T}_{12} \end{bmatrix}, \quad \boldsymbol{T}_{12} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & x_1 & y_1 \\ 0 & x_2 & y_2 \end{bmatrix}$$

for general transversal displacement functions w has to be respected. To calculate the mass effects for the displacement w, the reformulation of the form functions

$$\boldsymbol{g} = \underbrace{\boldsymbol{g}^k + \frac{1}{2}\boldsymbol{V}^T \begin{bmatrix} 2r^2 s + (3r - s + t)rst \\ 2s^2 t + (3s - t + r)rst \\ 2t^2 r + (3t - r + s)rst \end{bmatrix}}_{=\boldsymbol{g}^0} + \underbrace{\frac{3}{2}\boldsymbol{V}^T \operatorname{diag}\left(\begin{bmatrix} \mu_3 \\ \mu_1 \\ \mu_2 \end{bmatrix}\right)}_{=\boldsymbol{V}} \underbrace{\begin{bmatrix} (r + s - t)rst \\ (s + t - r)rst \\ (t + r - s)rst \end{bmatrix}}_{=\boldsymbol{g}^1} \tag{18}$$

is more convenient. Exact integration using eq. (12) implies

$$\psi(\boldsymbol{g}^{\boldsymbol{0}} \cdot \boldsymbol{g}^{\boldsymbol{0}T}) = \frac{1}{100800} \begin{bmatrix} 10096 & -2228 & 1114 & 3352 & 626 & -986 & 3352 & 626 & 360 \\ -2228 & 544 & -272 & -986 & -178 & 288 & -986 & -178 & -110 \\ 1114 & -272 & 163 & 626 & 117 & -178 & 360 & 61 & 49 \\ 3352 & -986 & 626 & 10096 & 1114 & -2228 & 3352 & 360 & 626 \\ 626 & -178 & 117 & 1114 & 163 & -272 & 360 & 49 & 61 \\ -986 & 288 & -178 & -2228 & -272 & 544 & -986 & -110 & -178 \\ 3352 & -986 & 360 & 3352 & 360 & -986 & 10096 & 1114 & 1114 \\ 626 & -178 & 61 & 360 & 49 & -110 & 1114 & 163 & 109 \\ 360 & -110 & 49 & 626 & 61 & -178 & 1114 & 109 & 163 \end{bmatrix},$$

$$\psi(\boldsymbol{g}^{\boldsymbol{1}} \cdot \boldsymbol{g}^{\boldsymbol{0}T}) = \frac{1}{151200} \begin{bmatrix} 194 & -52 & 30 & 194 & 30 & -52 & 32 & 8 & 8 \\ 32 & -16 & 8 & 194 & 22 & -52 & 194 & 22 & 30 \\ 194 & -52 & 22 & 32 & 8 & -16 & 194 & 30 & 22 \end{bmatrix},$$

$$\psi(\boldsymbol{g}^{\boldsymbol{1}} \cdot \boldsymbol{g}^{\boldsymbol{1}T}) = \frac{1}{75600} \begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{bmatrix}.$$

This leads to the implicit equation for the mass matrix

$$\int_{OP_1P_2} \delta w \cdot w\, dA = \Delta\, \boldsymbol{\delta\gamma}_w{}^T \boldsymbol{T_3}^T \left(\psi(\boldsymbol{g}^0 \boldsymbol{g}^{0T}) + \boldsymbol{G}^{10} + \boldsymbol{G}^{10T}\right) \boldsymbol{T}_3^w \boldsymbol{\gamma}_w$$

$$\text{with} \quad \boldsymbol{G}^{10} = \boldsymbol{V}^{\mu} \left(\psi(\boldsymbol{g}^1 \boldsymbol{g}^{0T}) + \frac{1}{2}\psi(\boldsymbol{g}^1 \boldsymbol{g}^{1T}) \boldsymbol{V}^{\mu T}\right)$$

due to the transverse deformation w. In [11], example calculations have shown that the calculated stiffness coefficients are too high if exact integration is applied to the variational integral for potential energy. A reduced integration at the special points with barycentric coordinates $2/3$, $1/6$, and $1/6$, which is exact up to order two, was the most accurate evaluation method and is also applied here. Using the matrices

$$\boldsymbol{G}^{uv} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}, \quad \boldsymbol{G}_0^w = \frac{1}{6} \begin{bmatrix} -26 & 19 & -1 & -2 & 1 & 1 & 28 & 6 & 2 \\ -32 & 16 & -8 & 16 & 8 & -2 & 16 & 8 & -6 \\ -32 & 16 & 9 & -14 & 3 & 7 & 46 & 13 & -2 \\ 16 & -2 & 8 & -32 & -8 & 16 & 16 & -6 & 8 \\ -2 & 1 & 1 & -26 & -1 & 19 & 28 & 2 & 6 \\ -14 & 7 & 3 & -32 & 9 & 16 & 46 & -2 & 13 \\ 28 & -8 & 2 & -2 & -2 & 1 & -26 & -18 & -1 \\ -2 & 1 & -2 & 28 & 2 & -8 & -26 & -1 & -18 \\ 10 & -5 & 6 & 10 & 6 & -5 & -20 & -11 & -11 \end{bmatrix}, \quad \boldsymbol{G}_1^w = \frac{1}{9} \begin{bmatrix} -5 & 4 & -2 \\ -8 & 4 & -8 \\ -11 & 4 & -5 \\ -8 & -8 & 4 \\ -5 & -2 & 4 \\ -11 & -5 & 4 \\ 4 & -5 & -2 \\ 4 & -2 & -5 \\ 4 & 1 & 1 \end{bmatrix}$$

as abbreviations, the strain tensor at these three dedicated Gauss points is given by

$$\begin{bmatrix} \boldsymbol{\varepsilon}(\frac{2}{3}, \frac{1}{6}, \frac{1}{6}) \\ \boldsymbol{\varepsilon}(\frac{1}{6}, \frac{2}{3}, \frac{1}{6}) \\ \boldsymbol{\varepsilon}(\frac{1}{6}, \frac{1}{6}, \frac{2}{3}) \end{bmatrix} = \boldsymbol{D}_3^u \boldsymbol{G}^{uv} \boldsymbol{\gamma}_u + \boldsymbol{D}_3^v \boldsymbol{G}^{uv} \boldsymbol{\gamma}_v - z \boldsymbol{D}_3^w (\boldsymbol{G}_0^w + \boldsymbol{G}_1^w \boldsymbol{V}^{\mu\, T}) \boldsymbol{T}_3^w \boldsymbol{\gamma}_w.$$

A tedious calculation proves this strain tensor representation by insertion of the deformations u, v, and w into eq. (2) and a suitable reordering of terms. The matrices to transform the derivatives with respect to (r, s) into strain components

$$\boldsymbol{D}_3^u = \begin{bmatrix} \boldsymbol{D}_{12}^u \\ \boldsymbol{D}_{12}^u \\ \boldsymbol{D}_{12}^u \end{bmatrix}, \quad \boldsymbol{D}_3^v = \begin{bmatrix} \boldsymbol{D}_{12}^v \\ \boldsymbol{D}_{12}^v \\ \boldsymbol{D}_{12}^v \end{bmatrix}, \quad \boldsymbol{D}_3^w = \begin{bmatrix} \boldsymbol{D}_{12}^w \\ \boldsymbol{D}_{12}^w \\ \boldsymbol{D}_{12}^w \end{bmatrix}$$

are stacked matrices, obtained from eq. (8). The stiffness matrix is now implicitly given by the quadratic form

$$\frac{\Delta}{6} \int_{z_m - h/2}^{z_m + h/2} \begin{bmatrix} \boldsymbol{\delta\varepsilon}(\frac{2}{3}, \frac{1}{6}, \frac{1}{6}) \\ \boldsymbol{\delta\varepsilon}(\frac{1}{6}, \frac{2}{3}, \frac{1}{6}) \\ \boldsymbol{\delta\varepsilon}(\frac{1}{6}, \frac{1}{6}, \frac{2}{3}) \end{bmatrix}^T \begin{bmatrix} \boldsymbol{C} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{C} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{C} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varepsilon}(\frac{2}{3}, \frac{1}{6}, \frac{1}{6}) \\ \boldsymbol{\varepsilon}(\frac{1}{6}, \frac{2}{3}, \frac{1}{6}) \\ \boldsymbol{\varepsilon}(\frac{1}{6}, \frac{1}{6}, \frac{2}{3}) \end{bmatrix} dz.$$

The factor $1/6$ represents the equal weights of the quadrature formula. The integration with respect to z is evaluated using eq. (13).

The piezoelectric coupling vector $\boldsymbol{f_p}$ has to be calculated with eq. (7). Side integrals with linearly varying functions f can be evaluated by

$$\int_{\boldsymbol{p}_1}^{\boldsymbol{p}_2} f \, ds = \frac{\|\boldsymbol{p}_2 - \boldsymbol{p}_1\|}{2} (f(\boldsymbol{p}_1) + f(\boldsymbol{p}_2)).$$

A combination of all three sides leads to

$$\int_{\partial OP_1P_2} f \, \boldsymbol{n} \, ds = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix} \boldsymbol{Z}^{uv} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \quad \text{with} \quad \boldsymbol{Z}^{uv} = \frac{1}{2} \begin{bmatrix} \boldsymbol{p}_1 & \boldsymbol{p}_2 \end{bmatrix} \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \end{bmatrix},$$

which will be used for the membrane part of eq. (7). To evaluate the part depending on the transversal displacements w, the normal derivatives

$$\frac{\partial f}{\partial \boldsymbol{n}_j} = \frac{s_j}{2\Delta} \begin{bmatrix} 1 \\ \mu_j \end{bmatrix}^T \boldsymbol{N}_j^{\boldsymbol{n}} \begin{bmatrix} f,_r \\ f,_s \end{bmatrix} \quad \text{with} \quad \boldsymbol{N}_1^{\boldsymbol{n}} = \begin{bmatrix} -2 & 1 \\ 0 & 1 \end{bmatrix}, \; \boldsymbol{N}_2^{\boldsymbol{n}} = \begin{bmatrix} 1 & -2 \\ -1 & 0 \end{bmatrix}, \; \boldsymbol{N}_3^{\boldsymbol{n}} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (19)$$

are needed. The validity of these relationships can be checked by substitution of (15) into (11). Altogether, this results in

$$\int_{\partial OP_1P_2} \frac{\partial f}{\partial \boldsymbol{n}}\, ds = \frac{1}{4\Delta} \begin{bmatrix} {s_1}^2 \\ {s_2}^2 \\ {s_3}^2 \end{bmatrix}^T \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & -2 & 1+\mu_1 & 0 & -2 & 1+\mu_1 \\ 0 & 1-\mu_2 & -2 & 0 & 0 & 0 & 0 & 1-\mu_2 & -2 \\ 0 & 1+\mu_3 & 1-\mu_3 & 0 & 1+\mu_3 & 1-\mu_3 & 0 & 0 & 0 \end{bmatrix}}_{=\boldsymbol{Z}^w} \boldsymbol{T}_3^w \boldsymbol{\gamma}_w,$$

the boundary integral of the normal derivative of $w = \boldsymbol{g}^T \boldsymbol{T}_3^w \boldsymbol{\gamma}_w$. The coupling vector $\boldsymbol{f_p}$ is finally given implicitly by the variational relationship

$$\delta \boldsymbol{q}^T \boldsymbol{f_p} = e_{31}^\star \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}^T \boldsymbol{Z}^{uv} \delta \boldsymbol{\gamma}_u + \begin{bmatrix} -1 \\ 0 \end{bmatrix}^T \boldsymbol{Z}^{uv} \delta \boldsymbol{\gamma}_v - z_m \frac{1}{4\Delta} \begin{bmatrix} {s_1}^2 \\ {s_2}^2 \\ {s_3}^2 \end{bmatrix}^T \boldsymbol{Z}^w \boldsymbol{T}_3^w \delta \boldsymbol{\gamma}_w \right).$$

In [5] the Specht's incompatible triangular element as well as the Clough Tocher macro element with piezoelectric coupling is discussed more extensively. Detailed derivations of the equations, calculation of the coefficient matrices with the computer algebra system MATHEMATICA©, validation code, an element implementation in MATLAB© and application of the patch test is done there.

4.6 Piezo Patch System Equations

If the constitutive eq. (3) are extended by the direct piezoeffect, the force equations contained implicitly in (6) are finally extended to the augmented discrete system of differential equations

$$\hat{\boldsymbol{M}}_{\boldsymbol{p}} \ddot{\boldsymbol{u}}_{\boldsymbol{m}} + \hat{\boldsymbol{K}}_{\boldsymbol{p}} \boldsymbol{u}_{\boldsymbol{m}} - \hat{\boldsymbol{f}}_{\boldsymbol{p}} U = \hat{\boldsymbol{f}}_{\boldsymbol{m}}, \tag{20}$$

$$\hat{\boldsymbol{f}}_{\boldsymbol{p}}{}^T \boldsymbol{u}_{\boldsymbol{m}} + C_p U = Q \tag{21}$$

for the considered piezoceramic module. The coupling vector $\hat{\boldsymbol{f}}_{\boldsymbol{p}}$ occurs in the so-called actuator eq. (20) and also in the sensor eq. (21) due to energy considerations. $\boldsymbol{u}_{\boldsymbol{m}}$ denotes the mechanical states, $\hat{\boldsymbol{f}}_{\boldsymbol{m}}$ the external forces, U the electric potential supplied to the piezo electrodes,

$$C_p = \frac{\epsilon_{33}^\star A}{h} \quad \text{with} \quad \epsilon_{33}^\star = \epsilon_{33}^\sigma - \frac{2E}{1-\nu} {d_{31}}^2$$

the capacity and Q the external charge transfered from the electrical network, the piezo patch is attached to.

5 Modal Coupling Approach

The modal coupling approach is the central idea to develop a fast method to augment elastic components of an elasto-kinematic system with active components. Although

the equations obtained so far can all be finally assembled by a classical finite element program, there are some arguments to have more flexible and faster ways to position the adaptronic devices. Firstly, it is comfortable to be able to enhance exported finite element models with actuators in other environments. Secondly, a fast repositioning of these devices is crucial to the problem of finding optimized locations of several devices with respect to some objective function. Control laws are also designed in matrix orientated environments like MATLAB©, therefore the established model must be exported from the finite element program anyway. Finally, the assembly of the elastic system at different points of the trajectory is more convenient in external environments too.

5.1 Modal Decomposition of the Underlying Substructure

In order to use the modal coupling approach, first the system (1) should be decoupled by a modal analysis in the finite element program. The eigenfrequencies and corresponding eigenmodes, which are most important to the simulation goals for a certain application as visualized in Fig. 5 must be selected and exported. It is advisable to export some more modes as the ones, which are contained in the selected frequency range, because the modal coupling approach is an approximation, converging to the true solution as the number of respected modes tend to infinity. It is sufficient to only export that part of the mesh covering the area, where piezo patches might be attached and to export possibly additional nodes, which are used to join neighbouring components of the mechanism. Corresponding to eq. (1) this leads to the modal description

$$\boldsymbol{M}\,\ddot{\boldsymbol{q}}_m + \boldsymbol{D}\,\dot{\boldsymbol{q}}_m + \boldsymbol{K}\,\boldsymbol{q}_m = \boldsymbol{f}_m$$

with $\boldsymbol{M} = \boldsymbol{\Phi}^T \hat{\boldsymbol{M}} \boldsymbol{\Phi}$, $\boldsymbol{D} = \boldsymbol{\Phi}^T \hat{\boldsymbol{D}} \boldsymbol{\Phi}$, $\boldsymbol{K} = \boldsymbol{\Phi}^T \hat{\boldsymbol{K}} \boldsymbol{\Phi}$, $\boldsymbol{f}_m = \boldsymbol{\Phi}^T \hat{\boldsymbol{f}}$ and $\boldsymbol{u} = \boldsymbol{\Phi}\,\boldsymbol{q}_m$. The colums of $\boldsymbol{\Phi}$ represent the exported eigenmodes. The modal mass and stiffness matrices $\boldsymbol{M}$ and $\boldsymbol{K}$ are diagonal matrices. Their diagonal elements are known as modal mass and stiffness coefficients. If the modal masses are normed to be unity, the modal stiffnesses are the squares of the natural frequencies $\omega_j = 2\pi f_j$. If modal damping

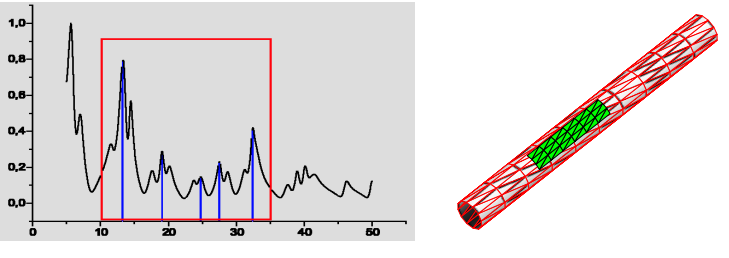


Fig. 5. a Principle selection of important mode shapes for the application. **b** Applied patches need mesh interpolation strategies to adapt DOF's.

is assumed, the matrix $\boldsymbol{D}$ is also of diagonal shape and the structural equations are fully decoupled.

5.2 Augmenting the Structural Model with Patch Actuators

If some patch actuators should be attached to an elastic component, the principle idea is to assemble the system of equations of the augmented structure in modal space. Therefore, each patch will add correction matrices $\boldsymbol{M}_p = \boldsymbol{\Phi}^T \hat{\boldsymbol{M}}_p \boldsymbol{\Phi}$ and $\boldsymbol{K}_p = \boldsymbol{\Phi}^T \hat{\boldsymbol{K}}_p \boldsymbol{\Phi}$ to the original matrices $\boldsymbol{M}$ and $\boldsymbol{K}$. The coupling vectors $\boldsymbol{f}_p = \boldsymbol{\Phi}^T \hat{\boldsymbol{f}}_p$ are also transformed to modal space. If all these corrections are done, the fully coupled system

$$\boldsymbol{M}_a \ddot{\boldsymbol{q}}_m + \boldsymbol{D}\, \dot{\boldsymbol{q}}_m + \boldsymbol{K}_a\, \boldsymbol{q}_m = \boldsymbol{f}_m + \boldsymbol{F}_p \boldsymbol{U},$$
$$\boldsymbol{F}_p{}^T \boldsymbol{q}_m + \boldsymbol{C}_p \boldsymbol{U} = \boldsymbol{Q}$$

is obtained. The second equation is vectorized to emphasize the use of multiple applied ceramics. Each ceramic has its own triangle mesh as shown in Fig. 5. To be able to position the ceramic patches at arbitrary locations on the exported mesh, the DOF's of the piezo mesh have to be interpolated to the DOF's of the exported structural mesh. For this purpose, fast universal mesh interpolation routines have been developed as described in [12].

6 A Simple Example

As a simple example of the theory presented so far, the planar model of an active slider-crank mechanism is considered (see Fig. 6). Though this example is very simple, it can be used to study some of the aspects, covered by the theory of active elastic bodies in multibody scenarios. A rigid crank rotates with a fixed angular velocity ω. The axis of rotation is fixed in the origin of the inertia system. The slider has a known mass and is movable on the x-axis. The link between the rotating end of the slider and the center of the slider mass is given by an elastic beam, whose flexibility is described by the first three bending modes. The structural model of the beam has simple support on both ends. The Euler-Bernoulli beam theory can

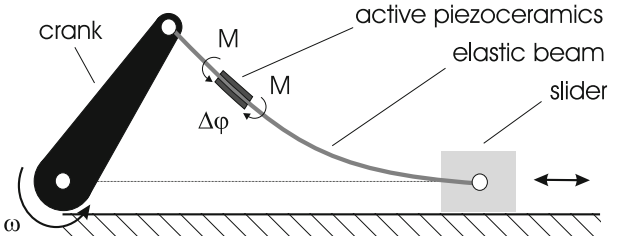


Fig. 6. A simple slider crank mechanism with one patch actuator.

Table 1. Parameters for the slider-crank model

Description	Symbol	Value
length of crank	l_c	300 mm
length of beam	l_b	500 mm
angular velocity of crank	ω_c	25.13 rad s^{-1}
Youngs modulus of beam	E_b	$7 \cdot 10^{10}$ N m^{-2}
mass of slider	m_s	40 g
mass density of beam	ρ_b	2700 kg m^{-3}
width×height of beam	$w_b \times h_b$	30 mm × 2 mm

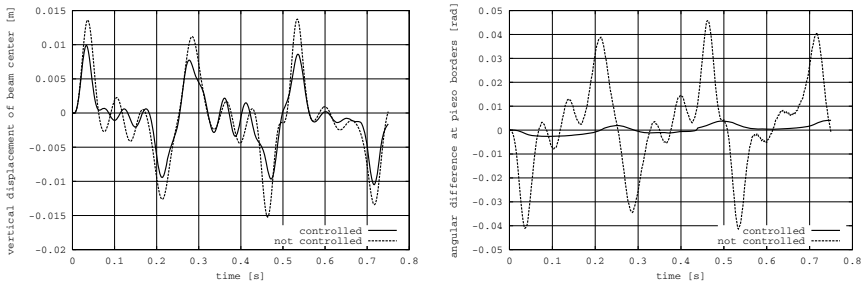


Fig. 7. a Uncontrolled and controlled deformation of the beam. **b** Uncontrolled and controlled bending of the piezoelements.

be used, because the beam is sufficiently flat. Geometric stiffening effects are important and included in the simulation. The gravitational forces act in the negative y-axis. At some point (25 cm apart from the crank) of the beam, two collocated piezoceramic elements of length 10 cm are applied on both sides. The upper one is used as actor, and the lower one as sensor. Because of the collocated positions, the structure won't get unstable by a positive velocity feedback control [13]. The parameters for the numerical simulation are listed in Table 1. The beam is made of aluminium. In the simulation, the upper actuator is voltage driven. Therefore, only the actuator eq. (20) is needed. The lower sensor is connected to a charge amplifier. Therefore, zero voltage is supplied, and the amplified charges represent the sensor signal. Only the sensor eq. (21) is needed in this case. By using the simple control law $U_a = K \cdot \dot{Q}_s$, for a suitable choosen control gain K, the results presented in Fig. 7 are obtained. The control effect of the piezoceramic on the global deformation of the beam is present but not very impressive. This is due to the fact that in this example the input signal for the controller is not closely related to the bending modes. Nevertheless, the controller does a very good job in decreasing the sensor signal.

7 Conclusion

Light weight robot structures allow short cycle times in assembly tasks, but they show a tendency of undesirable structural vibrations due to low material damping. Adaptronic conceps use smart devices, like piezo ceramic patch actuators, in combination with suitable control algorithms to reduce these vibrations. This paper presented a new Finite Element, which can be used as a general electro-mechanical coupling element for such flat actuators. With the modal correction method, a flexible application and repositioning of these coupling devices is possible without time consuming recalculation of large Finite Element models. This fast coupling method requires the export of modal data (eigenfrequencies, damping coefficients, eigenmodes) from the structural model. Its approximative nature can be controlled by the number and selection of the eigenmodes. Triangular meshes and the eigenmode deformations are only needed for the appropriate surface parts where the application of the piezo ceramic patches is possible or desirable.

Control strategies, especially with respect to the non-linear dependency of the structural model of the robot with respect to the location of the end-effector in the workspace, are discussed elsewhere [9]. The choice of good actuator and sensor locations is still a general topic of active research in the adaptronic community. In [14] a solution based on genetic algorithms is presented for acoustic problems.

Acknowledgements. The author gratefully acknowledges the support of this work by the German Research Foundation (DFG) within the Collaborative Research Centre SFB 562 “Robots for Handling and Assembly — Highly Dynamic Parallel Structures with Adaptronic Components”.

References

1. Zienkiewicz, O., Taylor, R.: The Finite Element Method. Basic formulations and linear problems, vol. 1. McGraw-Hill, London (1989)
2. Tzou, H.: Piezoelectric Shells. Kluwer Academic Publishers, Dordrecht (1993)
3. Zemcik, R., Rolfes, R., Rose, M., Teßmer, J.: High-performance four-node shell element with piezoelectric coupling for the analysis of smart laminated structures. International Journal for Numerical Methods in Engineering 70(8), 934–961 (2007)
4. Mesecke-Rischmann, S.: Modellierung von flachen piezoelektrischen Schalen mit zuverlässigen finiten Elementen, Dissertation, Helmut Schmidt Universität, Hamburg, Germany (2004)
5. Rose, M.: Modale Korrekturmethoden für die Platzierung von Piezokeramischen Modulen. Technical Report IB 131-2004/43, German Aerospace Center (DLR) (2004)
6. Dietz, S.: Vibration and fatigue analysis of vehicle systems using component modes. In: Fortschritt-Berichte VDI, vol. 12, pp. 1–136. VDI Verlag, Düsseldorf (1999)
7. Clark, R., Saunders, W., Gibbs, G.: Adaptive Structures. John Wiley & Sons, New York (1998)
8. Preumont, A.: Vibration Control of Active Structures. Kluwer Academic Publishers, London (1997)

9. Algermissen, S., Sinapius, M.: Robust Gain Scheduling for Smart-Structures in Parallel Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 159–174. Springer, Heidelberg (2010)
10. Argyris, H.: Die Methode der Finiten Elemente, Germany,, vol. I. Friedr. Vieweg & Sohn, Braunschweig (1986)
11. Specht, B.: Modified shape functions for the three-node plate bending element passing the patch test. International Journal for Numerical Methods in Engineering 26, 705–715 (1988)
12. Rose, M.: An advanced branch and bound method to interpolate acoustic data on structural finite element meshes. In: 13th Congress on Sound and Vibration ICSV, Vienna, Austria (2006)
13. Kelkar, A., Joshi, S.: Control of Nonlinear Multibody Flexible Space Structures. Springer, London (1996)
14. Heintze, O., Rose, M., Algermissen, S., Misol, M.: Development and experimental application of a pre-design tool for active noise and vibration reduction systems. In: Prof. of the International Symposium on Active Control of Sound and Vibration, Ottawa, Ontario, Canada, pp. 1–12 (2009)

Design and Implementation of Adaptronic Robot Components

Ralf Keimer and Michael Sinapius

Abstract. Parallel robots demonstrated their potential in applications with the needs for high dynamic trajectories in recent years [1]. In contrast to serial robots they meet the demands for higher accelerations at constant precision due to their fixed drives and large structural stiffness. On the other hand higher accelerations increase the risk of severe vibration of the structural parts. Therefore smart structures technology is introduced in parallel robotics for a further enhancement of the features of parallel robots. In consequence, fundamental investigations regarding adaptive mechanical components and adaptive systems are carried out in order to extend the application range of parallel robots. The development of specific active components being able to reduce the vibrations are investigated within the field of smart structures for parallel robots and are elucidated in the article. Manufacturing aspects and design principles are addressed.

1 Introduction

Short cycle times are crucial in handling and assembly. Short cycle times require high acceleration and deceleration, which in turn may induce structural vibrations. Standard robot control is mainly coping with rigid-body motion of kinematics and the structural vibrations are usually disregarded, resulting in either a reduced accuracy on the trajectory and especially at the end of the trajectory, or in longer cycle-times, when waiting until structural damping reduces the vibration before starting the next task.

In order to sustain high accuracy while still offering short cycle-times reduction of vibration is addressed by influencing the structural dynamics of the robot to suppress the disturbance.

Ralf Keimer · Michael Sinapius
German Aerospace Center (DLR), Institute of Composite Structures and Adaptive Systems, Lilienthalplatz 7, 38108 Braunschweig, Germany
e-mail: {ralf.keimer,michael.sinapius}@dlr.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 413–427.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

The structural dynamics of the robots can actively be influenced by three different measures:

- Using the drives of the robot,
- Influencing the joints of the structure [2, 3, 4],
- Integrating active components like active struts in bar members of the structure [5, 6].

The latter are designated as adaptronic robotic components as they

- do not interfere with the positioning and control of the kinematics (which is mostly seen as addressing rigid body motion) in opposite with using the drives, where rigid-body control and vibration suppression act on the same actuator,
- are less complex than joints, which concentrate complex mechanical elements in a small amount of space (like axles, bearings),
- offer themselves as possibility to integrate further functions beside their load carrying function as vibration suppression.

The core of adaptronics or smart structures technologies[1] is to influence the elasto-mechanical structural behavior by means of the integration of actuators and sensors directly into the structural load path. Adaptive components are based on multi-functional materials being able to carry loads and having simultaneously actuating capabilities which are based on energy conversion [7, 8]. These capabilities can be utilized for shape control of the structure built by smart material since the material converts energy being electrical, thermal, optical or magnetic into mechanic deformation. Whenever the energy conversion is quick enough it can be used for dynamic purposes as well. Active vibration control (AVC) is a main area of application [9]. AVC means the active suppression of structural vibrations through the chain of measuring the vibration, determination of active measures in fast controllers and counteraction of the undesirable vibrations through actuators built with smart materials.

2 Multi-functional Materials

Numerous smart materials have been developed and investigated during the past decades which are characterized by their multifunctionality. Multi-functionality means that these materials are energy converters and are able to carry loads. This property makes them appropriate for being integrated into the load path of structures or structural components. The most advanced material which is already widely used is piezoceramic, mainly PZT.

The energy conversion capability of PZT is based on a polarized lattice. At temperatures below the so-called Curie temperature, strain can be evoked either by external loads or by an electrical field which deforms the lattice due to its dipole character. Figure 1 depicts an piezoceramic actuator reacting to the applied

[1] In Europe the keyword “Adaptronics" is more common, in America and Asia "Smart structures" is usually used.

Fig. 1. Piezoceramic actuation principle

electrical field generated by the voltage U, where Π is the direction of polarization and thus the orientation of the lattice dipoles.

More generally the strain ϵ can be expressed by the simplified linear relation

$$\epsilon_k = s_{kl}^E \sigma_l + d_{ik} E_i \tag{1}$$

which mathematically elucidates the main feature of smart materials, i.e. being sensitive to external mechanical loads σ_l and an electrical field E_i. s_{kl} and d_{ik} are the coefficients of proportionality, i.e. the flexibility and the piezoelectric coupling coefficient, respectively. The indices i, k, l denominate directions with respect to the orientation of the lattice, where i is the direction of polarization, k the direction of strain to be calculated, and l the direction of stress. For actuation purposes the transversal and longitudinal piezoceramic effects are used, i.e. an electrical field is applied along the axis of polarization and the elongation perpendicular to the direction of the electrical field or the elongation parallel to the field are used respectivley. The piezoelectric coupling and flexibility for the longitudinal setup is greater than for transversal, resulting in having approximately twice work capacity.

However, the direct use of smart materials as construction material is impossible in most cases due to material characteristics like brittleness, so that special care has to be taken when designing adaptive robot components.

3 Structural Integration and Structural Conformity

Structural integration of smart materials in the structural load path requires the consideration of structural conformity, i.e. the optimization of actuation and load carrying capability. This is elucidated by a simple example. Figure 2 shows a piezoceramic actuator having a stiffness c_a embedded in a composite having the stiffness c_c. The stiffness relation between the smart material and the composite carrying the loads is essential for the usable actuation. The load dependent deflection of the actuator is

$$u = u_0 + \frac{P(u)}{c_a} \tag{2}$$

where u_0 is the unimpeded stroke of the piezoceramic, c_a the stiffness of the actuator and $P(u)$ the load.

The elasticity of the structural attachment determines the acting force, which is given by

$$P(u) = -c_c u \tag{3}$$

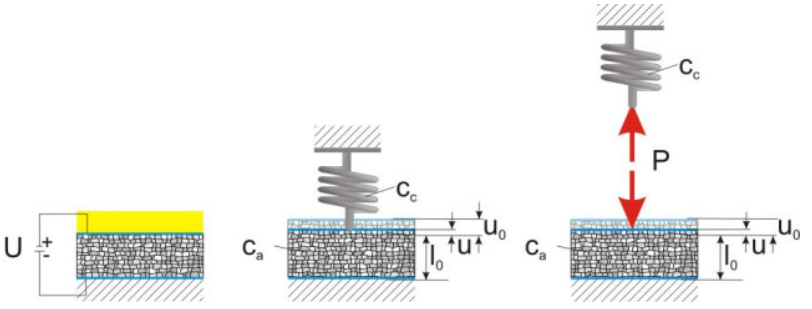


Fig. 2. Embedded smart material

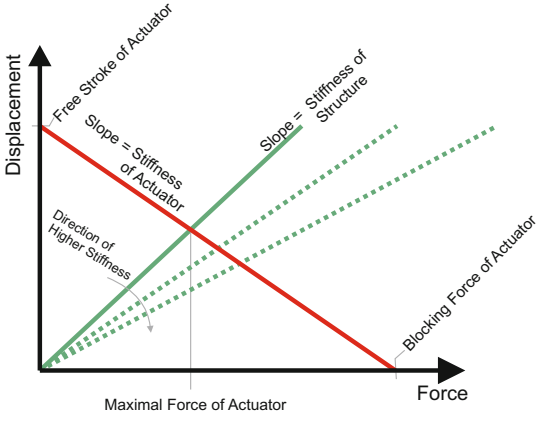


Fig. 3. Work diagramm of actuator with respect to stiffness of structure

where c_c is the stiffness of the structure. Figure 3 depicts this acting force as intersection between the working diagramm of the actuator and the stiffness of the structure.

Introducing the stiffness relation between the smart layer and the embedding structure

$$\lambda = \frac{c_c}{c_a} \tag{4}$$

the actuator stroke can be written as

$$u = u_0 \frac{1}{1+\lambda} \tag{5}$$

The applied work amounts to

$$\begin{aligned} w_a &= \frac{1}{2} c_c u^2 \\ &= \frac{1}{2} \frac{\lambda}{(1+\lambda)^2} u_0^2 c_a \end{aligned} \tag{6}$$

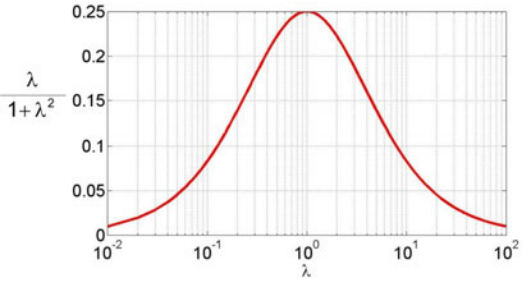


Fig. 4. Work capacity of embedded smart material

expressed with the actuator-parameters free stroke u_0 and stiffness c_a. Figure 4 illustrates this relation.

The coefficient $\frac{\lambda}{(1+\lambda)^2}$ expresses the influence of the stiffness relation between the smart layer and the embedding structure on the maximum working capacity of the active composite. For $\lambda = 1$ a maximum of structural conformity is realized in this very simple case which is characterized by an unidirectional load carrying requirement. Structural conformity is a measure for active structural components, i.e. a measure for adaptation capability of the structure.

However, the realization of structural conformity is a central engineering process of adaptronics since load carrying capability includes all possible load cases like

- stiffness including buckling
- strength
- actuation stroke
- work capacity

The engineering process of realization of structural conformity is elucidated below for active robot components.

4 Vibration Suppression through Destructive Interference

Different concepts of active vibration control (AVC) based on smart materials have been developed in recent years [10, 11, 12]. They may be distinguished via their control concepts (feedback or feed forward control) and/or the target of the vibration suppression. There are two principle approaches to actively suppress vibration. The first principle targets the modal control of a structure by actively reducing the amplitudes of structural modes. Thus this AVC approach is 'global'. The second approach targets the active control of structural waves whenever the flow of vibrational energy from one structural part to another is important. This approach leads to the tranquillization of defined structural points. It should be noted that this approach may lead to an increase of vibrational amplitudes in other structural parts. Hence, this approach of AVC is 'local'. However, both approaches are based on the principle of destructive interference which is elucidated in Fig 5.

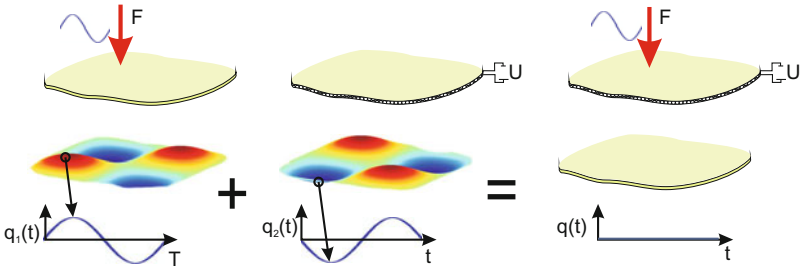


Fig. 5. Principle of Destructive Interference

AVC can be realized optimally trough integrated actuators since they can be located in the load path. Thus they have an optimal authority to control the structural vibration. A second option of active vibration control is the application of vibration absorbers which mistune the structural resonances. Tunable absorbers are able to adjust their operation frequency to varying environmental conditions. Hence, they are in particular suited to effectively suppress vibration in systems with varying dynamic properties.

In the case of robotics the approach of vibration absorbers is not recommendable, due to variying loads and changing structural dynamics. Using integrated actuators good results can be achieved, using the appropiate controller [13].

5 Structural Integrable Actuators

Due to the industrial availability, the well known behaviour and the relatively low prices piezoceramic actuators are used to integrate into robotic structures. Two kinds of piezoactuators can be distinguished, stack type actuators and foil actuators (Fig 6).

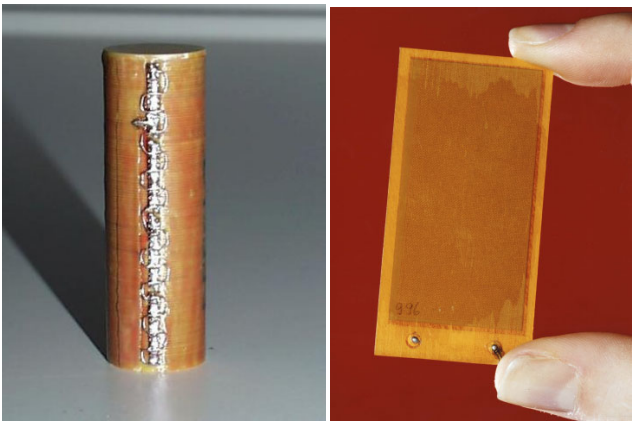


Fig. 6. Piezoactuators, left: Stack Actuator, right: Foil Actuator

Their different properties lead to different designs regarding the aforementioned structural conformity.

5.1 Stack Actuators

Stack actuators as shown on the left side in Fig 6 are a common configuration of piezoceramic actuators. Alternating layers of contacting material and piezoceramics are stacked to build the actuator. This can be done on the one hand using ready sintered piezoceramic foils with metallic electrodes resulting in high voltage stacks usually driven with 1000 V, and on the other hand by producing monolithic stacks where the electrical contacts are produced in the process of sintering the ceramic material. In both cases stack type actuators are using the longitudinal piezoelectric effect.

Electrical contacting is inherent in the stack actuator and electrical insulation is easily added. However prestressing for the needed structural conformity has to be designed into the structure, where this kind of actuator is to be embedded.

5.2 Foil Actuators

Pure piezoceramic foils are less readily usable as contacting and insulation as well as prestressing has to be provided. Nonetheless research within the last years [14, 15] led to ready packaged actuators with simultaneous electrical contacting, electrical insulation, mechanical stabilization, and prestressing. A typical configuration is depicted in Fig 7. Most of the foil actuators use the transversal piezo-effect, but some of them are built using the longitudinal effect. These semi-finished products are ready to be applied on structural surfaces through a proper bonding or structural integration into composite structures. A main advantage of foil actuators compared with stack type actuators is, that prestressing is realized through proper packaging.

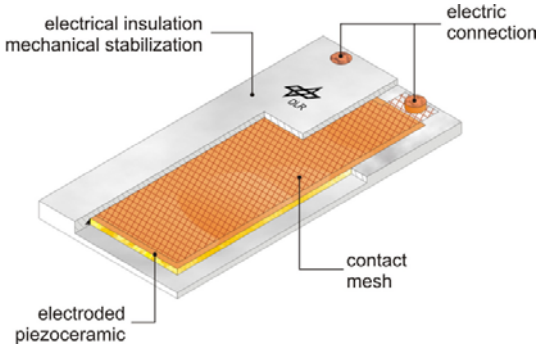


Fig. 7. Piezocomposite with integrated insulation, electrical contact and prestressing

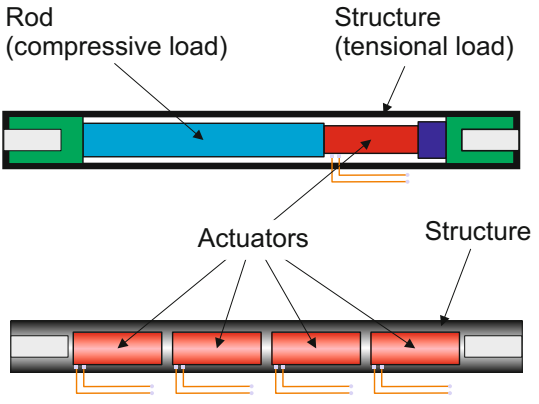


Fig. 8. Topologies of active rods, top: Rod with discrete piezostack, bottom: Rod with distributed surface actuators

5.3 Active Rods as Example for Structural Integration

Both kinds of actuators, i.e. stack actuators and foil actuators, have been investigated for the application in parallel robotics consisting of rods as load carrying elements. The upper part of Fig 8 depicts a setup with a discrete piezostack. Prestressing is needed in order to counteract the brittleness of the piezoceramic and enabling positive and negative displacements of the actuator. Prestressing causes a tensional loaded structure and a compressed rod, assuming that the stack doesn't expand to the full length of the rod, which is a sensible assumption considering the requirement for low moved masses. The related configuration [16, 5] is optimized with respect to optimum work capacity. The topology of the related configuration (upper part of Fig 8) is based on a more complex design than the one shown at the lower part of the figure, where thin actuators are attached to the surface of the rod's structure. The stiffness relation λ from equation 4 is higher than for the discrete stack actuator. On the other hand the free stroke u_0 is higher, as the actuators expand over the full length of the rod. In order to get the required prestressing, piezocomposites as shown in Fig 7 are used, which have an inherent prestressing.

6 Challenges and Requirements for Adaptive Robot Components

The advantage of parallel robots is the chance to realize higher accelerations at constant precision compared with serial robots since they do not move the driving units. In consequence the structure supporting the end effector has to be as light as possible in order to realize the targeted high accelerations. On the other hand a maximum stiffness is required for ensuring highest precision of the end effectors position. However, extreme light weight designs increase the risk of vibration of the structural parts due to high deceleration of the end effector which counteracts

the target of low cycle times. Introducing active measures for vibration reduction supports the goal to realize high dynamic trajectories.

The requirements for adaptive robot components can be summarized as follows:

- Extreme light weight construction of the structural components
- Low complexity, low actuator voltage
- Maximum structural conformance of integrated actuators

7 Design Principles for Optimal Structural Conformity

From the vibration suppression point of view, the goal of the design process is to maximize actuation stroke in order to give optimal authority for control, as the actuators are not operating against a global stiffness in general. In consequence, high forces are not required, but high actuation strokes are essential to effectively reduce vibration with destructive interference. The actuation stroke u of a bar member is given in equation 5. In order to reach high strokes two parameters can be adapted:

- The stiffness ratio λ, e.g. by maximising cross-section of actuating material and thus reach a low λ.
- The free stroke u_0, e.g. by maximising the length of the actuator, as u_0 is determined by the length of the actuator and the active strain of the actuating material.

While there is no global stiffness, actuators integrated into the structure have to work against local stiffnesses of the structures building the robot components, because usually these components are not solely built out of actuators. Maximisation of actuation stroke is not the sole target of the design process. Structural conformity is a second important goal [17]. Structural conformity is defined by three characteristics:

- The active member has to be able to carry the external loads, resulting from the robot tasks.
- The active member has to be in conformance with the requirement of low moved masses in the robot,
- The actuator of the component has to be able to carry the internal loads. Since piezoceramic material is brittle, a prestressing of the actuator is important for proper operation, as vibration suppression results in alternating dynamic loads.

Different topologies for active rods are possible depending on the chosen actuator as is depicted in Fig 8. The topology for active rods using patch-actuators is simpler than the one using piezostacks, as the prestressing of the ceramic material is inherent in the setup of patch-actuators as depicted in Fig 7 [15, 14]. The design process realizing structural conformity is much easier using patch-actuators. Moreover, the overall design is much more simple.

The combination of patch-actuators with fibre reinforced plastics in structures enables a flexible dimensioning of active members, since the fibre orientation can be used to adapt the stiffness of the structure according to the loads while still being able to optimize the stroke of the active member.

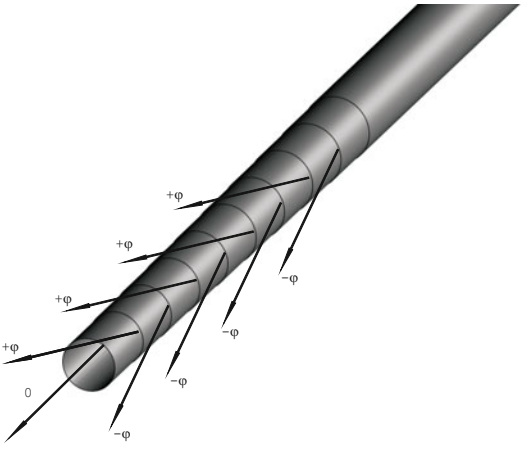


Fig. 9. Carbon fibre composite layout as structure of a rod

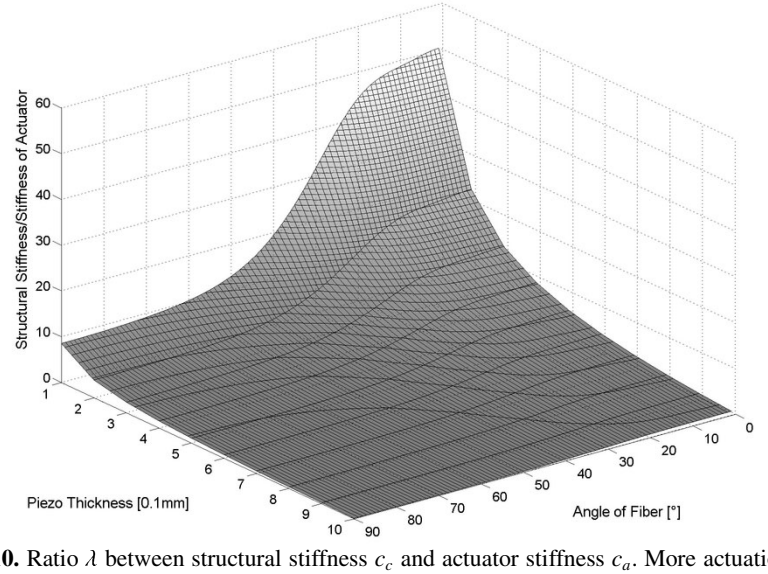


Fig. 10. Ratio λ between structural stiffness c_c and actuator stiffness c_a. More actuation authority for lowest ratio λ

As example, Fig 9 depicts the fibre layout of a rod with a length of 42.4 cm, a radius of 1 cm and wall-thickness of 2 mm. The rod is used to demonstrate the dimensioning procedure [18]. The ratio λ depends on the thickness of the used actuators and the angle of the fibre-orientation φ, as is depicted in Fig 10. An optimum is reached for high angles of fibre-orientation combined with high thicknesses of piezoactuators, respectively. Two failure modes exist for a given dynamic load (2500 N in this example) against which the rods have to be dimensioned. On the

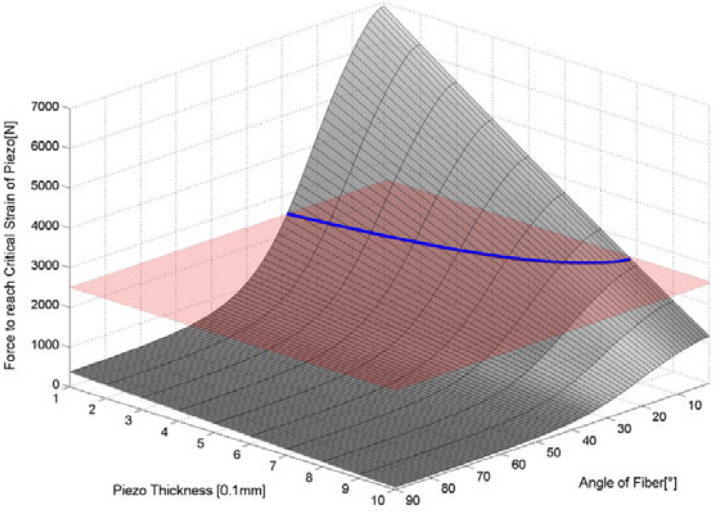


Fig. 11. Force needed to reach the critical strain of the embedded piezoactuator. The shaded plane shows the dimensioning load.

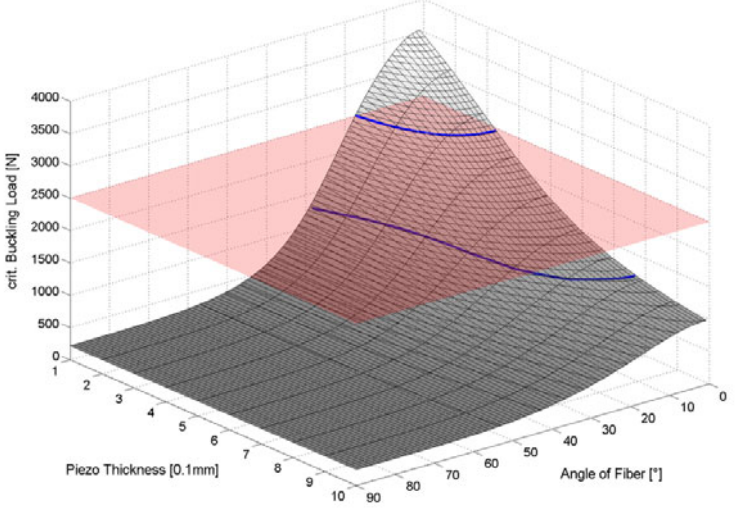


Fig. 12. Buckling load of the rod. The shaded plane shows the dimensioning load. Lower line shows limit from the critical strain.

one hand a tensional load leads to a critical strain of the embedded actuators which has to be avoided, as is shown by the shaded plain in Fig 11. On the other hand compressional loads have to be below the buckling load of the rod. This leads to limitations regarding the ratio λ due to these loads, because the parameters have

to be chosen in a way that bearable forces are above the dimensioning load. This is illustrated in Fig 12 where parameters above the shaded plain comply with this requirement. Of course matters of manufacturability can further reduce the field of possible parameters.

8 Manufacturing Aspects

The field of possible parameters identified in the previous chapter is purely defined by the working loads, always assuming that they can be realized while manufacturing the structure.

Availibilty of actuators in the desired dimensions and of course the question whether it is possible to apply the actuators on the carrying structure without damage are additional constraints to the design of adaptive robot components. A good synopsis of available foil actuators is given in [19].

In the example discussed in the previous section the ability to apply thick ceramics to a circular rod of small radius is limited due to reaching the critical strain of the flat embedded actuators while bonding. Tests have shown that bonding of actuators of the kind described in Fig 7 with 0.2 mm ceramic-layer can be attached to a rod of 15 mm diameter, while actuators with next thicker layer 0.5 mm ceramic are destroyed in the process of applying.

The consideration of these constraints leads to a thickness of the actutator of 0.2 mm and an angle of 15° for the orientation of the fibre. Using this values a rod with optimal structural conformity can be build.

9 Adaptive Robot Components

Depending on the used kinematics in parallel robots, members have to carry longitudinal loads (rods), bending loads (beams) or complex loads including torsional loads. Different kinds of actuated structural components can be designed using the

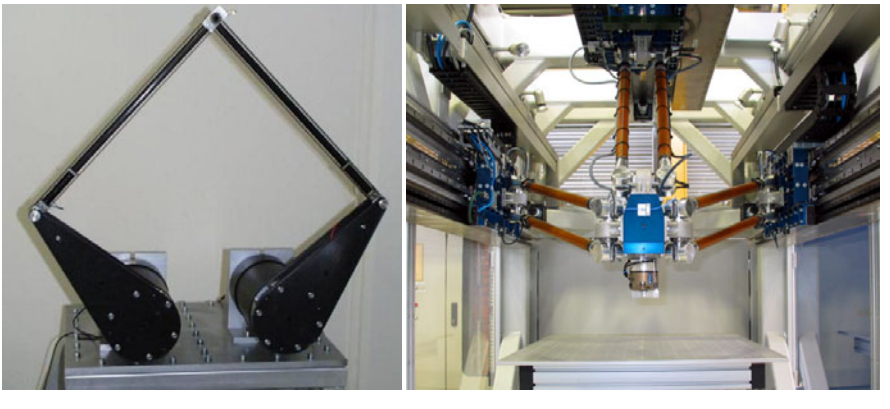


Fig. 13. Robots, left: FIVE-BAR, right: TRIGLIDE.

Fig. 14. Rod for FIVE-BAR

Fig. 15. Rods for TRIGLIDE, left: single patch-actuator and rod, right: complete link built of two rods.

concept of structural conformity. Exemplary structures are shown in Fig 13. The corresponding active members are presented in Fig 14 and Fig 15, each representing one type of active rod as is shown in Fig 8. The rod depicted in Fig 14 is built for the FIVE-BAR robot using a piezostack-actuator. The dimensioning load was assumed to be pure longitudinal. Prestressing is realized using an outer tension belt. The setup is quite simple, but dimensioning against more complex loads as additional torsional or bending loads would lead to a different design regarding every part of this concept. The second type of rod is depicted in Fig 15. It is dimensioned exactly as it is elucidated in the previous section, even with combined loads,

i.e. bending and torsional loads, as well as longitudinal loads. These combined loads are specified for the parallel robot TRIGLIDE.

10 Summary

The principle of structural conformity is presented as a central design rule for the design of adaptronic robot components used for active vibration suppression.

The examples demonstrate that the concept of surface mounted actuators is more general and more modular regarding the complexity to realise different setups. Especially dimensioning against combined loads is easier than with discrete piezostack actuators. Moreover, combined actuation like bending and longitudinal motion can be realized with patch-actuators as is shown in [6].

References

1. Schütz, D., Budde, C., Raatz, A., Hesselbach, J.: Parallel Kinematic Structures of SFB 562. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 109–124. Springer, Heidelberg (2010)
2. Pavlović, N., Keimer, R., Franke, H.-J.: Systematic development of adaptronic joints for parallel kinematic structures. In: Proceddings of 8th International IFAC Symposium on Robot Control, Bologna, Italy (2006)
3. Pavlović, N., Keimer, R., Franke, H.-J.: Design of an adapronic swivel joint for parallel robots based on high-frequency excitation. In: Proceedings of Joint Conferece on Robotics ISR 2006 / Robotik 2006, München, Germany (2006)
4. Pavlović, N., Keimer, R., Franke, H.-J.: Design of an adaptronic swivel joint for parallel robots based on quasi-statical clearance adjustment. In: Proceedings of 5th Chemnitzer Parallel Kinematics Seminar, Parallel Kinematic Machines in Research and Practice, pp. 341–355. Verlag Wissenschaftliche Scripten, Zwickau (2006)
5. Algermissen, S., Keimer, R., Rose, M., Breitbach, E.: High-speed parallel robots with integrated vibration-suppression for handling and assembly. In: Proceedings of SPIE-Smart Structures/NDE, San Diego, California, USA, SPIE 2004, SPIE-Smart Structures/NDE, San Diego, California, USA (2004)
6. Breitbach, E., Algermissen, S., Keimer, R., Rose, M., Stachera, C.: Adaptive tools in parallel robotics. In: Last, P., Budde, C., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly, 2nd International Colloquium of the Collaborative Research Center 562. Fortschritte in der Robotik, vol. 9, pp. 203–219. Shaker Verlag, Braunschweig (2005)
7. Janocha, H.: Adaptronics and Smart Structures. Springer, Heidelberg (1999)
8. Srinivasan, A.V., McFarland, D.M.: Smart Structures: Analysis and Design. Cambridge University Press, Cambridge (2001)
9. Clark, R.L., Saunders, W.R., Gibbs, G.P.: Adaptive Structures: Dynamics and Control. John Wiley & Sons, Inc., Chichester (1998)
10. Fuller, C.R., Elliott, S.J., Nelson, P.A.: Active control of vibration. Acad. Press, London (1997)
11. Préumont, A.: Vibration Control of Active Structures: An Introduction. Solid Mechanics And Its Application. Kluwer Academic Publishers, Dordrecht (1997)
12. Gawronski, W.K.: Dynamics and Control of Structures: A Modal Approach. Mechanical engineering series. Springer, Heidelberg (1998)

13. Algermissen, S., Sinapius, M.: Robust Gain Scheduling for Smart-Structures in Parallel Robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 159–174. Springer, Heidelberg (2010)
14. Wierach, P., Sachau, D.: Design and manufactoring of complex adaptive structures with piezoceramic patch actuators. In: Proceedings of International Mechanical Engineering Congress and Exposition, New York, USA (2001)
15. Wierach, P., Monner, H.P., Schönecker, A., Dürr, J.K.: Application specific design of adaptive structures with piezoceramic patch actuators. In: Proceedings of SPIE's 9th Annual Symposium on Smart Structures and Materials, San Diego, USA (2002)
16. Breitbach, E., Keimer, R., Rose, M., Sachau, D.: An adaptronic solution to increase efficiency of high speed parallel robots. In: Proceedings of 12th International Conference on Adaptive Structures and Technologies (ICAST), College Park, Maryland, USA (2001)
17. Keimer, R., Algermissen, S., Pavlovic, N., Budde, C.: Smart structures technologies for parallel kinematics in handling and assembly. In: Proceedings of SPIE - Smart Structures & Materials/NDE, San Diego, California, USA (2007)
18. Algermissen, S., Keimer, R., Rose, M., Monner, H.P.: Robust control for vibration suppression on parallel robot triglide. In: Proceedings of Adaptronic Congress, Göttingen (2006)
19. Wierach, P., Schönecker, A.: Bauweisen und Anwendungen von Piezokompositen in der Adaptronik. In: Proceedings of Adaptronic Congress, Göttingen, Germany (2005)

Passive and Adaptive Joints for Parallel Robots

Nenad Pavlović, Robert Otremba, David Inkermann
Hans-Joachim Franke, and Thomas Vietor

Abstract. In order to take advantage of parallel kinematic structures, the development of specific and optimized structure components (rods, passive joints) is becoming increasingly necessary. Accordingly, in this chapter the development and implementation of novel joint concepts for parallel structures is discussed. The focus lies on two different joint types: firstly, conventional passive joints and secondly, adaptive joints with integrated piezo-actuators. Based on the specific design methodology, different joint prototypes are developed and tested. By drawing on the test results, it is shown how the novel joints influence performance of the whole robot system.

1 Introduction

Parallel kinematic structures feature some significant advantages compared to serial mechanisms. Their drives are mainly mounted on the fixed rack, which is reflected in low moving masses and in high velocity and acceleration. Exploiting the advantages in dynamics and stiffness, some parallel structures have already been applied in different industrial areas: DELTA-structure [1] for handling and assembly, TRICEPT-structure [2] for manufacturing, Stewart- [3] and Gough-Platform [4] for motion simulation etc.

Despite the aforementioned advantages, this kind of mechanism did not become established in industrial applications. The reason for that is its numerous drawbacks: occurrence of singularities within the workspace [5], poor ratio of the workspace to the installation space, complex control and lack of standard structure components. Different strategies for overcoming the drawbacks and accentuating the advantages of parallel kinematic structures are therefore needed.

Nenad Pavlović · Robert Otremba · David Inkermann ·
Hans-Joachim Franke · Thomas Vietor
Technische Universität Braunschweig, Institute for Engineering Design,
Langer Kamp 8, 38106 Braunschweig, Germany
e-mail: {pavlovic,otremba,inkermann,franke,vietor}@ikt.tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 429–444.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

The first and the most important step in structure development is a systematic structure synthesis [6, 7]. Additionally, methods for structure modularization play an important role, in order to make synthesis process more flexible as well as to offer further alternative structure concepts [8, 9]. Some advanced approaches also include structure development in relation to the appropriate application (task) [10, 9] and/or to appropriate requirements [11]. In order to take advantage of the structure, various parameters (e.g. crank length, rod length, rack radius, working platform radius) have to be optimized. Correlations between the parameters are mostly complex (non-linear) and not explicit, so that a multi-criterion optimization [12, 13] provides the only possible solution.

In the early days of development of parallel kinematic structures, it was attempted to build them by using standard market components. Very soon it became clear that in this way it was very difficult to exploit their advantages (especially in dynamics and stiffness) and to obtain better performances than with serial mechanisms. The focus is therefore shifting more and more to the development of specific and optimized structure components, both in industry and in scientific circles. In this context, the most important components are rods and joints [14, 15]. Rods connect drives to working platform, transmit loads between them and enable movement of the working platform. Rods can assume some other functions beside the basic ones. For example, it is possible to integrate linear or rotational drives in the rods [16] in order to change their movability. In this way, innovative structure design as well as novel control strategies can be developed [15, 17]. Additionally, rods can be used for vibration suppression in the structure. Two different methods are currently well-established. The first one is based on embedding of multifunctional materials into CFK rods. Such adaptive rods can oscillate (due to embedded piezoceramic actuators) against the operational excitation and suppress vibrations in the structure [18]. The second method is based on integration of specific adaptronic modules with piezoceramic actuators in conventional rods. The adaptronic module increases the torsional stiffness of the rods and at the same time reduce their sensitivity to vibrations [19].

This chapter focuses on the development of joints for parallel robots. Joints are very important components, which join other structure components to one other, enable relative movement between them and influence the behavior of the whole robot system. It is shown that by means of specific and optimized joints, some of the disadvantages of parallel structures can be overcome. Two different joint types are discussed here: firstly, conventional passive joints and secondly, adaptive joints with integrated piezo-actuators.

2 Passive Joints for Parallel Robots

Passive joints which are available on the market (e.g. from INA-Schaeffler KG, Hephaist SEIKO Co., Ltd.) have been primarily developed for parallel kinematic machine tools. Since they are big and heavy and feature relatively small swivel angles, their application in parallel robots is fairly limited. Other available joint

solutions were mostly developed in combination with corresponding structures, so that they are not applicable in general. In that context, some kind of general approach for development of joints for parallel robots becomes necessary.

The initial methodology for systematic development of passive joints for parallel robots was proposed by Otremba [20, 21]. The objectives of that work were gathering information about all major features of passive joints, then development and linking of systematic methods by which joints of parallel structures can be selected, modified or newly developed, and finally verification of the methodology by development and testing of the joint prototypes.

2.1 Classification System for Joint Selection

The kinematic configuration of the required joints is specified within the kinematic synthesis of a parallel mechanism. Based on the specified joint configuration, an adequate joint type has to be selected. In order to eliminate joints which are inappropriate for the given task (e.g. ball-and-socket-joints, if a large motion range is required), a complete design catalogue of theoretical joint configurations is required. In order to improve this process, development of the tools for the joint selection becomes necessary. In [22, 21] a suitable classification system for joint selection is presented (Table 1).

Table 1. Classification system for joint selection.

Classification Level	**Item**
I - External Kinematic Configuration	1. Degree of Freedom 2. Form of Relative Motion & Mutual Position of Joint Axes
II - Internal Kinematic Configuration	3. Internal Configuration of Joint Axes 4. Variation with Different Tools
III - Bearing Configuration	5. Configuration of Bearing Pairs 6. Movement Behavior at Contact Point

The classification system includes three levels:

(a) *External Kinematic Configuration.* This classification level includes two items: degree of freedom and form of relative motion & mutual position of joint axes. They are determined by the robot architecture, and all possible joint kinematic orders can be found by combining them. This general approach results many redundancies and nonsensical solutions. Most of them can be systematically eliminated by applying different rules. The result is a total of 18 different basic joint types with degrees of freedom between 1 and 5.

(b) *Internal Kinematic Configuration.* The external kinematic configuration can be realized internally by three different kinematic configuration types: integral, serial or parallel. The integral kinematic configuration is more suitable for joints

with smaller degrees of freedom and results in small size, simple design and high stiffness. Serial configurations are more suitable for joints with more than one degree of freedom and feature a higher movability. Joints based on this concept are generally bigger because they consist of two or more "sub-joints". Parallel kinematic configurations are less common joint concepts. They have many drawbacks, but for some problems they provide the only possible solution [23]. In order to find further possible configurations, different tools (e.g. catalogues) can be used. This results in further variations (e.g. inversion of hollow and full joint elements), which can influence different joint characteristics (e.g. size, movability, stiffness).

(c) *Bearing Configuration.* The bearing configuration is specified by the configuration of the bearing pairs and the movement behavior at the contact points. Bearing pairs describe a movable interconnection between two rigid bodies. They can be designed as mono- or poly-joints, i.e. they consist of one or more surface pairs. For each of the surface pairs, it is necessary to define a movement behavior at the appropriate contact point. Generally, there are three different possibilities: rolling, sliding or levitation pairs. Most of the conventional joints designed are based on the rolling pairs.

By classifying all known joints with detailed data and feasible design types, it is possible to reduce significantly the variety of possible solutions. Furthermore, it is possible to return to the structural synthesis, if appropriate joints are unavailable or unfeasible.

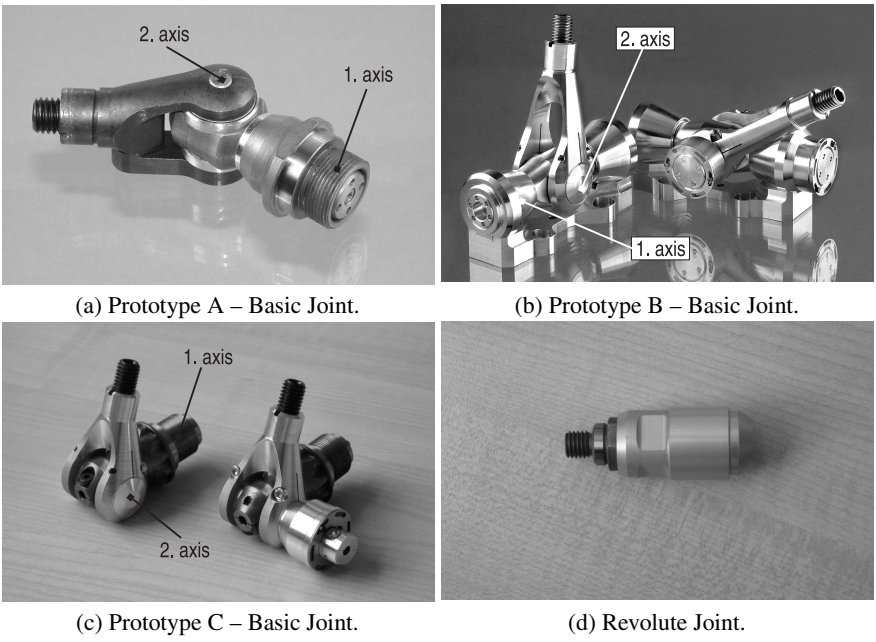
2.2 Joint Prototypes

Based on the presented classification system and using methods presented in [21], joint prototypes with one or more degrees of freedom are developed. They are tested both in the test bench and in the appropriate parallel robots: FIVE-BAR-structure [24, 25], HEXA-structure [26, 21, 25] and TRIGLIDE-structure [18, 27, 25]. In this section, three different joint prototypes for the HEXA parallel robot and the corresponding test results are presented.

2.2.1 Joint Prototypes for HEXA Parallel Robot

Joint prototypes developed for the HEXA parallel robot are designed in modular form and have three rotational degrees of freedom. They consist of a basic joint with two rotational degrees of freedom (axes are perpendicular to each other – serial configuration, Fig. 1 (a)-(c)) and can be expanded to joints with three rotational degrees of freedom. The module for the third axis is a revolute joint, this axis being perpendicular to the second axis of the basic joint (Fig. 1(d)).

By minor design changes of the joints, angle sensors can be integrated (Fig. 1(b) and 1(c)). Based on the data obtained from the integrated sensors, solving of the direct kinematic problem becomes much easier and some additional functions (e.g. online calibration) become feasible [28, 29].



(a) Prototype A – Basic Joint. (b) Prototype B – Basic Joint.

(c) Prototype C – Basic Joint. (d) Revolute Joint.

Fig. 1. Joint prototypes for HEXA parallel robot.

Prototype A - basic joint (Fig. 1(a)) has two axes, which are perpendicular to each other. The swivel angle for the first axis is ±360° and for the second one ±133°. The bearing of the first axis is one-sided and realized by a single radial needle roller bearing and a pair of axial needle roller bearings. The fork consists of two pieces and therefore is not symmetrical.

Compared to the prototype A, prototype B (Fig. 1(b)) features some differences. The bearing of the first axis is two-sided, which is reflected in a bigger size and higher weight. The swivel angle for the first axis is ±90° (50 % compared to the prototype A) and for the second one ±60° (less than 50 % compared to the prototype A). Furthermore the fork is one-piece and symmetrical.

Prototype C (Fig. 1(c)) is to a certain extent a mixture of the first two prototypes and includes "good" features of both of them. The bearing of the first axis is designed one-sided with two radial needle roller bearings and a pair of axial needle roller bearings. The result is the same motion range (first axis: ±360°, second axis: ±133°) but a slightly larger envelope compared to prototype A. The fork geometry is adopted from the prototype B (one-piece) and the joint is symmetrical.

2.2.2 Experimental Investigation

In order to assess performance of the developed joint prototypes, their load-carrying capacity as well as their stiffness were experimentally investigated [21, 26]. In this

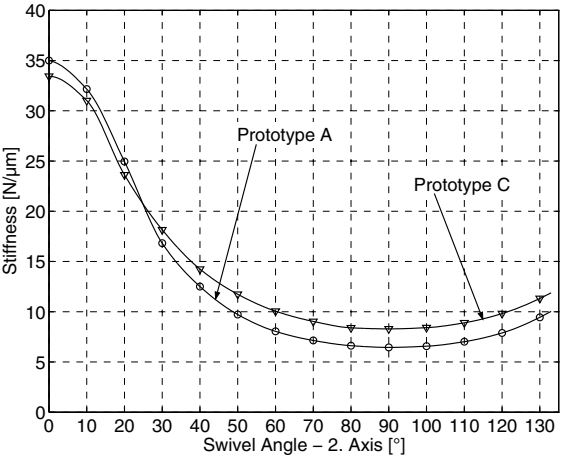


Fig. 2. Stiffness characteristics for joint prototypes A and C.

section, comparison of the stiffness characteristics for all three joint prototypes is discussed.

Methodology

Joint stiffness was investigated with an in-house test bench. The first joint axis was fixed, the joints were tensile-loaded on the fork and the resulting deformation was measured. The stiffness was calculated from the data colected. The joints were investigated in different positions:

- prototypes A and C: swivel angle of the first axis is irrelevant (joint is symmetrical relative to the first axis) and the swivel angle of the second axis varied between 10°-130°
- prototype B: swivel angle of the first axis varied between 0° and 90° and the swivel angle of the second axis varied between 0° and 60°

Results

Test results for the joint prototypes A and C are shown in Fig. 2. It is noticeable that in the investigated range, joint stiffness decreases with the increase of the swivel angle of the second joint axis. At smaller swivel angles, the dominant load for the first joint axis is tension while at bigger angles bending becomes dominant. Since the shaft (first axis) in the prototype A is shorter (one radial bearing is integrated) than the shaft in the prototype C (two radial bearings are integrated side by side), the stiffness of the prototype A is higher in the range between 0° and 25°. When the swivel angle increases, bending becomes dominant and the bearing of the first axis significantly influences the joint stiffness. Therefore, in the useable angle range for the HEXA parallel robot (45°-130°), prototype C features higher stiffness (up to 30 %) and provides a better solution.

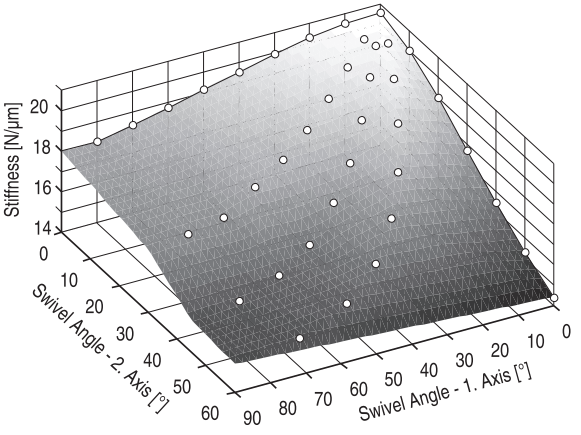


Fig. 3. Stiffness characteristic for joint prototype B.

Test results for the prototype B are shown in Fig. 3. The stiffness characteristic is more compact than the stiffness characteristics of the prototypes A and C: the difference between the maximum and the minimum stiffness is about 28 N/μm for the prototype A, about 25 N/μm for the prototype C and about 6 N/μm for the prototype B. So Prototype B has the better stiffness behavior due to its swivel angles and also a higher stiffness in absolute values, but has a slightly higher weight plus a larger envelope and smaller swivel angles. To find an optimal joint solution for the given robot structure, the requirements regarding movability, envelope, rigidity, weight and stiffness need to be balanced.

In different operation modes of parallel robots, requirements on joints are often different and contradictory. One of the most important aspects is the relation between clearance and friction in the joints: fast movement of the robot demands low friction in the joints, for slowing down high friction/damping is necessary and finally, during the assembly process, there should not be any clearance in the joints (because of the preferable accuracy). The goal conflict caused by these contradictory requirements is usually solved (also in the passive joints presented above) with compromise solutions with respect to required clearance and friction. Both requirements together are optimally matched, but it is not possible to achieve ”best friction” and ”best clearance” in the joint in different operation modes.

3 Adaptive Joints for Parallel Robots

In order to solve the aforementioned goal conflict, development of joints which can be adapted to different operation conditions becomes necessary. Generally, the joint adaptability can be achieved in two different ways: non-actively and actively.

Non-active joint adaptability is achieved by using bearings with an adjustable clearance (angular contact ball bearings, tapered roller bearings, angular contact spherical plain bearings etc.). Thanks to the clearance adjustment before/after the robot is in operation, it is possible to obtain the required friction performance. A

disadvantage of this solution is that it is only possible to adapt the joint while the robot is at a standstill.

In this section, adaptive joints for parallel robots which can actively match different operating conditions are discussed. Thanks to integration of components made of multi- functional materials (actuators), clearance and friction in the joints can be influenced and contradictory requirements can be fulfilled. Two different working principles are discussed: high-frequency excitation and quasi-statical clearance adjustment.

In the following, the design of laboratory revolute joint prototypes based on both working principles as well as corresponding results of the experimental investigations are presented.

3.1 Adaptive Joints Based on High-Frequency Excitation

This working principle is based on using actuators for generation of high-frequency oscillations. The oscillations cause friction reduction between joint components which are in contact and which are movable relative to each other. The required friction performance of the joint can be obtained by frequency/amplitude regulation. This effect is also experimentally investigated for linear motion [30, 31, 32].

Using this working principle and depending on the integrated bearings (axial and/or radial), a modular system of revolute joints can be developed [33]. In this section, a laboratory joint prototype with integrated radial and axial (one-sided) bearing is discussed.

3.1.1 Laboratory Joint Prototype

The design of the developed joint prototype is shown in Fig. 4. The applied actuator is a high-voltage (1000 V) piezo ring (1), with a length of 18 mm, an outside diameter of 35 mm and an inside diameter of 25 mm. In a no-load condition, it can achieve a maximum stroke between 17 and 25 μm at a stiffness of 1000 N/μm. The piezo actuator is pre-stressed between two flanges (2, 3). The necessary pre-stressing (piezo-ceramics must not be stressed on tension) is achieved by six screws (4), which are passed through centering bushings (5). The bushings enable centering of the flanges toward each other and positioning of the integrated bearings (6, 7) relative to the shaft (8).

The alternating voltage which is applied to the piezo ring causes its axial oscillations. In order to exploit them for the friction reduction, one of the two flanges has to be fixed. In this case, the lower flange is fixed to the test bench adapter (not shown) and the upper one oscillates. In the upper flange, standard radial (6) (inside diameter of 20 mm) and axial (7) plain bearings (inside diameter of 22 mm) are integrated. They oscillate together with the flange (2), move (axially) relatively to the shaft, and the friction between the oscillating bearings (6, 7) and the shaft (8) is reduced.

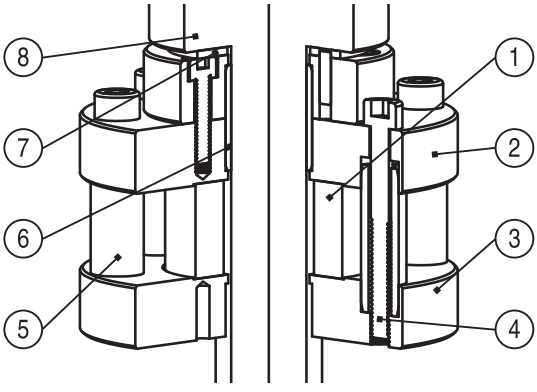


Fig. 4. Laboratory joint protoype based on high-frequency excitation.

3.1.2 Experimental Investigation

In order to verify the friction reduction in the joint, it was tested with an in-house test bench [34], which enables measurement of the friction in the joint prototype.

Methodology

The clearance in the radial direction corresponded to recommendations of the bearing manufacturer, and the axial plain bearing was lightly pre-stressed. The ratio of the macroscopic speed (shaft speed) to the vibration velocity amplitude (velocity generated by piezo actuator) varied from 0.5 to 1 for the radial bearing and from 0.75 to 1.5 for the axial bearing (maximum frequency was about 8 kHz). The velocity ratios for the radial/axial bearing are different because of the different radii where the friction takes place. For each velocity ratio, two tests were carried out: with and without using high-frequency excitation.

Results

Fig. 5 shows friction reduction in the joint prototype in relation to the ratio of the vibration velocity amplitude (generated by piezo actuator) to the macroscopic rotation speed (shaft speed).

The lower curve is determined computationally according to the mathematical model presented in [30, 32]. It shows friction reduction when the oscillating velocity is perpendicular to the macroscopic velocity, which is valid for both the axial and the radial bearing in the investigated prototype. Points on the left (rectangles) represent the experimentally determined friction reduction relative to the velocity ratio in the radial bearing. Points on the right (triangles) represent friction reduction relative to the velocity ratio in the axial bearing. The ”real” friction reduction is located between these two boundary cases. The trend lines correspond to the theoretically determined curve with a small deviation. The achieved friction reduction is about 20 %.

Beside the aforementioned positive effects, this solution features also some disadvantages. The most important ones are noise (especially at higher frequencies) and

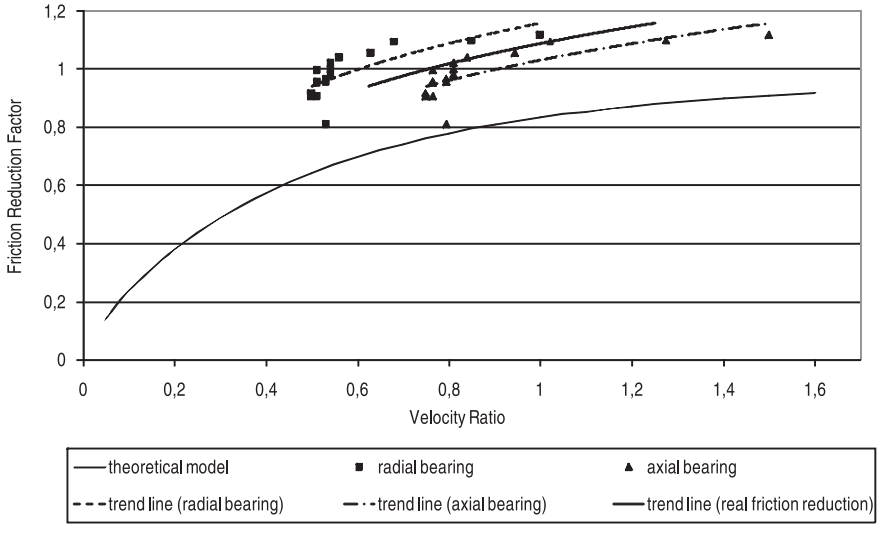


Fig. 5. Friction reduction in laboratory joint prototype based on high-frequency excitation.

necessity of powerful amplifiers for piezo actuators (in order to achieve a higher velocity ratio and therefore higher friction reduction).

3.2 Adaptive Joints Based on Quasi-statical Clearance Adjustment

This working principle is based on using actuators to influence the bearing assembly condition, in order to adjust the clearance for different operation conditions. Since the clearance is adjusted gradually and the actuators are used statically, the principle is called quasi-statical clearance adjustment. A similar principle had been proposed for the main spindle mounting of a lathe [35].

Based on the methodology presented in [36], the function structure for adaptive revolute joints based on quasi-statical clearance adjustment can be established [34]. Starting out from the function structure, and using different variation methods [37] and available design catalogues [38], different physical and corresponding working principles for each sub-function can be found. By combining them, different concepts of adaptronic joints can be developed [36]. Two of them shown as examples and will be discussed in the following.

3.2.1 Laboratory Joint Prototypes

In the laboratory joint prototypes shown in Fig. 6, clearance in the bearings is changed by activating piezoceramic actuators. The generated actuator force causes a higher normal force in the bearing surfaces and hence a higher friction. Using

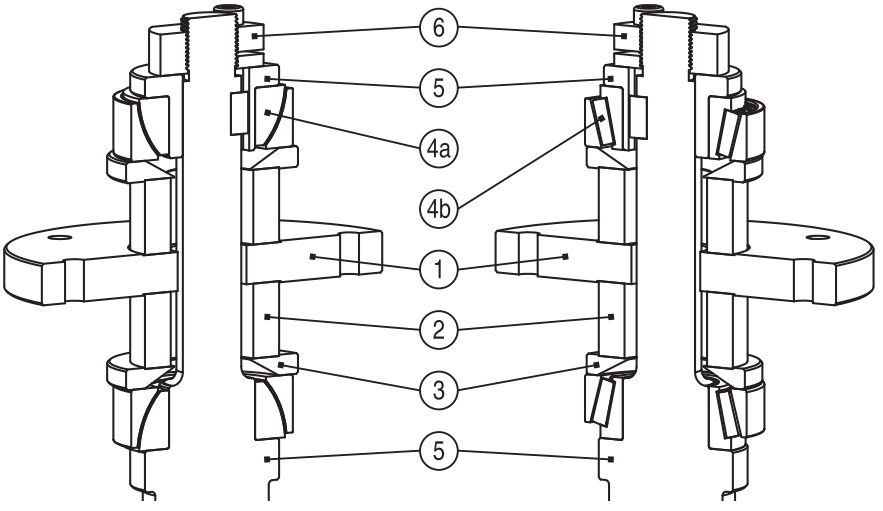


Fig. 6. Laboratory joint prototype based on quasi-statical clearance adjustment with integrated plain bearings (left) / rolling bearings (right).

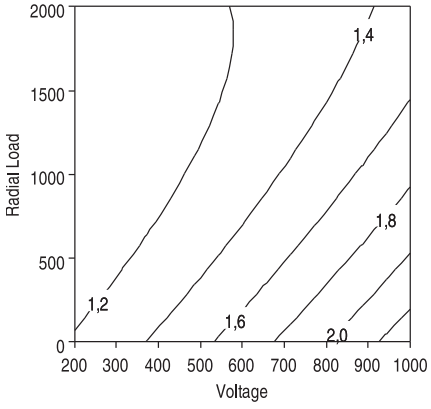
combined axial/radial bearings, clearance in both directions can be influenced simultaneously.

The basic part of the joint prototypes is a central ring (1), which enables integration of the joint in the test bench housing (not shown). On both sides of the ring, two high voltage (1000 V) piezo ring actuators (2) are glued. They have a length of 18 mm, outside diameter of 35 mm and inside diameter of 25 mm. In a no-load condition, they can achieve a maximum stroke between 17 and 25 μm at a stiffness of 1000 N/μm. Maximum force generation of the actuators is 20000 N. On the front surfaces of the piezo rings, two washers (3) are glued. They rest against the outer rings of the angular contact spherical plain bearings (4a), or rather tapered roller bearings (4b), which have a shaft diameter of 25 mm. The inner rings are axially fixed by shaft shoulders (5) and the required/desired pre-stressing of the bearings is achieved by a slotted round nut (6).

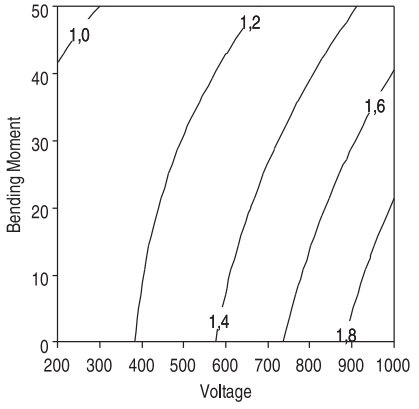
In the initial position there is a clearance between the bearing surfaces. When the piezoceramic actuators are activated, they shift the washers and the outer rings of the bearings. The inner rings of the bearings are blocked so that the clearance decreases. With deactivation of the actuators, the outer rings of the bearings return to their initial position and the predefined clearance is restored. In this way, different clearances can be adjusted and different friction moments achieved.

3.2.2 Experimental Investigation

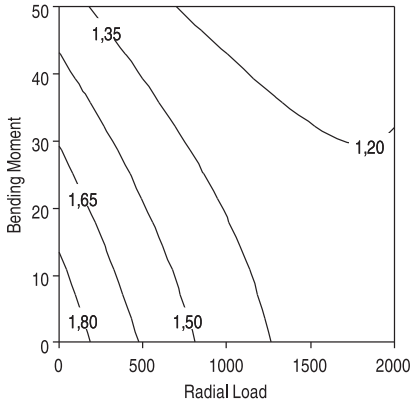
First experimental results of the laboratory joint prototype are presented in [36, 39, 34]. It is shown that the friction magnification as well as joint locking is possible by using piezoceramic actuators. In this work, correlation between friction



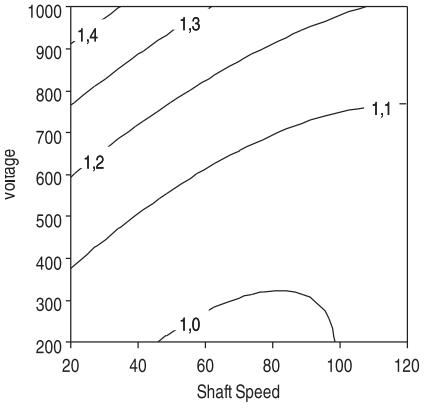
(a) Prototype with integrated plain bearings.



(b) Prototype with integrated plain bearings.



(c) Prototype with integrated plain bearings.



(d) Prototype with integrated rolling bearings.

Fig. 7. Friction magnification in laboratory joint prototypes based on quasi-statical clearance adjustment.

magnification and various parameters (shaft speed, applied voltage, axial load, radial load and bending moment) is discussed.

Methodology

The tests were carried out with an in-house test bench [34], which enables measurement of the friction moment in the investigated prototype. The speed profile was typical for parallel robots for handling and assembly (pick & place task) with speed values between 20 and 120 rpm. The applied voltage varied between 200 V and 1000 V. In order to simulate real operating conditions, the influence of different load types (axially, radially, bending) was investigated. The axial load was set to a range from 0 to 300 N, the radial load from 0 to 2000 N and the bending moment

from 0 to 50 Nm. The plain bearings were pre-stressed so that the friction moment in the unloaded joint was about 8 Nm (recommended value by bearing manufacturer is between 7 and 9 Nm) and rolling bearings were pre-stressed so that the friction moment corresponded to the recommendations of the bearing manufacturer (about 0.5 Nm).

Results

Test results for the laboratory joint prototype with plain bearings show that friction magnification (ratio between friction in the joint with and without using piezo-actuators) in the investigated range significantly depends on voltage, radial load and bending moment (Fig. 7, plots (a)-(c)). Dependence of the friction magnification on voltage is directly proportional (plots (a) and (b)). That is to be expected because the higher voltage induces a bigger actuator stroke and therefore a higher normal force in the bearings.

Dependence between radial load and friction magnification is reciprocal (plots (a) and (c)). This is caused by the friction characteristic of the integrated plain bearings. In this type of bearing, the friction increase caused by radial load is smaller at higher axial loads. Since the force induced by the actuators corresponds to an axial load for the bearings, the influence of the radial load falls off when the actuators are activated. The applied bending moment in the joint is mechanically equivalent to the radial load, so that its influence is also reciprocal (plots (b) and (c)). In the investigated range, radial load is more significantly than bending moment and has the same importance as the actuator.

Test results for the joint prototype with integrated rolling bearings show that the friction magnification in the investigated range significantly depends only on the shaft speed and on the applied voltage (Fig. 7, plot (d)). Dependence of the friction magnification on the applied voltage is directly proportional, similarly to the prototype with integrated plain bearings. On the other hand, dependence of the magnification on the shaft speed is reciprocal. Friction in the bearings increases at higher velocities, but the friction increase caused by the actuators is constant. In that way, the friction magnification is distinct at lower shaft speeds. The shaft speed and the applied voltage have approximately the same influence on the friction magnification.

Differences between the developed prototypes are also reflected in the test results. Depending on the requirements on the structure, it is possible to develop tailor-made joints for different applications. For example, if higher friction in joints is necessary because of the better damping, joints with plain bearings should be used. This joint type is also recommended if joint locking is necessary to enable functions like avoiding singularities, changing configurations, online calibration or high precision applications. Otherwise, whenever friction in joints should be as low as possible, it is recommended to use the joint concept with rolling bearings. In that case, it is necessary to have a separate (adaptronic) brake for eventual locking.

4 Conclusion

In this chapter, the development of passive joints for parallel structures is discussed. By using the highlighted methods, tools and novel physical principles, specific and optimized joints can be developed and the performance of parallel robots can be significantly improved.

The basic step in the joint development is the selection of a suitable kinematic configuration. In this work, a classification system for the joint selection which includes all possible joint configurations is proposed. The hierarchy of the classification system corresponds chronologically to the order in which requirements on the joints are available during the development process. Using the proposed classification system as well as other methods and tools, three joint prototypes are developed and tested. The realized prototypes are compact and light and feature higher movability and better stiffness characteristics than conventional joints which are commercially available.

Conventional passive joints provide a compromise solution according to contradictory requirements in different operation modes of the robot. In order to resolve this goal conflict, novel joint concepts (e.g. with integrated adaptronic components) are required. The presented laboratory joint prototypes, which are based on two different working principles (high-frequency excitation and quasi-statical clearance adjustment), allow adaptability of the clearance/friction characteristic while the robot is in operation. Development of the joints based on the presented concepts and their integration into real mechanisms will show how dynamic and damping performance as well as accuracy of parallel robots can be increased. All this will lead to shorter cycle times and therefore to a higher productivity.

Acknowledgements. The presented work was funded by the German Research Foundation (DFG) within the scope of the Collaborative Research Center SFB 562: Robotic Systems for Handling and Assembly.

References

1. Clavel, R.: Delta, a fast robot with parallel geometry. In: Proceedings of the 18th International Symposium on Industrial Robot, Lausanne, Switzerland, pp. 91–100 (1988)
2. Neumann, K.-E.: Exechon concept. In: Parallel Kinematic Machines in Research and Practice, Proceedings of the 5th Chemnitz Parallel Kinematics Seminar, Chemnitz, Germany, pp. 787–802 (2006)
3. Stewart, D.: A platform with 6 degrees of freedom. In: Proceedings of the IMechE, vol. 180, Pt.1(15), pp. 371–385 (1965)
4. Gough, V.E., Whitehall, S.G.: Universal tyre test machine. In: Proceedings of the 9th International Technical Congress of F.I.S.I.T.A., London, Great Britain, vol. 117, pp. 117–137 (1962)
5. Dietrich, F., Maaß, J., Bier, C., Pietsch, I., Raatz, A., Hesselbach, J.: Detection and Avoidance of Singularities in Parallel Kinematic Machines. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 77–92. Springer, Heidelberg (2010)

6. Hesselbach, J., Frindt, M.: Structural classification and systematic design of machines basing on parallel structures. In: Proceedings of the 31th International Symposium on robotics (ISR 2000), Montreal, Canada, pp. 65–70 (2000)
7. Frindt, M., Krefft, M., Hesselbach, J.: Structure and Type Synthesis of Parallel Manipulators. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 17–37. Springer, Heidelberg (2010)
8. Frindt, M.: Modulbasierte Synthese von Parallelstrukturen für Maschinen in der Produktionstechnik. Ph.D. thesis, TU Braunschweig. Vulkan Verlag, Essen (2001)
9. Stechert, C., Wrege, C., Franke, H.-J.: Task-based modular configurations for hybrid and redundant parallel robots. In: Proceedings of the 8th International IFAC Symposium on Robot control - SYROCO 2006, Bologna, Italy (2006)
10. Broghard, T., Hanssen, S., Hovland, G.: Application oriented development of parallel kinematic manipulators with large workspace. In: Last, P., Budde, C., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly, Proceedings of the 2nd International Colloquium of the Collaborative Research Center 562, Braunschweig, Germany, pp. 153–170 (2005)
11. Stechert, C., Franke, H.-J.: Requirement-oriented configuration of parallel robotic systems. In: Proceedings of the 17th CIRP Design Conference 2007 - The Future of Product Development, pp. 259–268. Springer, Berlin (2007)
12. Kirchner, J.: Mehrkriterielle Optimierung von Parallelkinematiken. Ph.D. thesis, TU Chemnitz. Verlag Wissenschaftliche Scripten, Zwickau (2001)
13. Krefft, M.: Aufgabenangepasste Optimierung von Parallelstrukturen für Maschinen in der Produktionstechnik. Ph.D. thesis, TU Braunschweig. Vulkan Verlag, Essen (2006)
14. Merlet, J.-P.: Parallel Robots. Springer, Dordrecht (2006)
15. Neugebauer, R. (ed.): Parallelkinematische Maschinen. Springer, Heidelberg (2006)
16. Wieland, F.: Entwicklungsplattform für Parallelkinematiken und Prototyp einer Werkzeugmaschine. Ph.D. thesis, TU Chemnitz. Verlag Wissenschaftliche Scripten, Zwickau (2000)
17. Tönshoff, H.K., Günther, G., Grendel, H.: Vergleichende Betrachtung paralleler und hybrider Strukturen. In: Neue Maschinenkonzepte mit parallelen Strukturen für Handhabung und Produktion. VDI Berichte, vol. 1427, pp. 249–270. VDI Verlag, Düsseldorf (1998)
18. Algermissen, S., Rose, M., Keimer, R., Monner, H.P., Sinapius, M.: Vibration control for smart parallel robots using robust gain-scheduling. In: ACTUATOR - International Conference on New Actuators, Bremen, Germany, pp. 429–432 (2008) ISSN ISBN3-933339-10-3
19. Wittstock, V.: Piezobasierte Aktor-Sensor-Einheiten zur uniaxialen Schwingungskompensation in Antriebstangen von Werkzeugmaschinen. Ph.D. thesis, TU Chemnitz. Verlag Wissenschaftliche Scripten, Zwickau (2007)
20. Franke, H.-J., Otremba, R., Jänicke, T.: Methodical development of optimized passive joints. In: Krefft, M., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly, Proceedings of the 1st International Colloquium of the Collaborative Research Center 562, Braunschweig, Germany, pp. 119–130 (2002)
21. Otremba, R.: Systematische Entwicklung von Gelenken für Parallelroboter. Ph.D. thesis, TU Braunschweig. Logos Verlag, Berlin (2005)
22. Otremba, R.: Ein Ordnungssystem zur gezielten Auswahl von Gelenken für kinematische Strukturen. In: Pini, P., Germer, C. (eds.) Konstruktionsmethodik in der Praxis – Einsatzmöglichkeiten und Grenzen, Braunschweig, Germany. Berichte des Instituts für Konstruktionstechnik, vol. 66, pp. 25–35 (2004)

23. Hamlin, G.J., Sanderson, A.C.: TETROBOT: A Modular Approach to Reconfigurable Parallel Robotics. Kluwer Academic Publishers, Boston (1997)
24. Algermissen, S., Keimer, R., Rose, M., Breitbach, E., Monner, H.P.: Applied robust control for vibration suppression in parallel robots. In: Proceedings of 22nd International Symposium on Automation and Robotics in Construction (ISARC), Ferrara, Italy (2005)
25. Schütz, D., Budde, C., Raatz, A., Hesselbach, J.: Parallel Kinematic Structures of the SFB 562. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 109–124. Springer, Heidelberg (2010)
26. Otremba, R.: Methodische Entwicklung von Gelenken für Parallelroboter. Konstruktion (10–2005), 82–86 (2005)
27. Budde, C.: Wechsel der Konfiguration zur Arbeitsraumvergrößerung bei Parallelrobotern. Ph.D. thesis, TU Braunschweig. Vulkan Verlag, Essen (2009)
28. Hesselbach, J., Bier,C., Pietsch, I, Plitea, N., Büttgenbach, S., Wogersien, A., Güttler, J.: Passive joint-sensors for parallel robots. Mechatronics (15) (2005)
29. Kirchhoff, M.R., Güttler, J., Wogersien, A., Pavlović, N., Otremba, R., Franke, H.J., Büttgenbach, S.: Design and Implementation of New Sensors and their Integration in Joints. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 445–459. Springer, Heidelberg (2010)
30. Littmann, W., Storck, H., Wallaschek, J.: Sliding friction in the presence of ultrasonic oscillations: superposition of longitudinal oscillations. Archive of Applied Mechanics 71, 549–554 (2001)
31. Liu, Y.: Reduzierung der Reibkraft durch Ultraschallwellen. Ph.D. thesis, Universität-GH Paderborn. Shaker Verlag, Aachen (1993)
32. Storck, H., Littmann, W., Wallaschek, J., Mracek, M.: The effect of friction reduction in presence of ultrasonic vibrations and its relevance to travelling wave ultrasonic motors. Ultrasonics 40(1), 379–383 (2002)
33. Pavlović, N., Keimer, R., Franke, H.-J.: Design of an adaptronic swivel joint for parallel robots based on high-frequency excitation. In: Proc. of the Joint Conferece on Robotics ISR 2006 / Robotik 2006, Munich, Germany (2006)
34. Pavlović, N., Keimer, R., Franke, H.-J.: Adaptronic revolute joints for parallel robots based on simultaneous quasi-statical axial nd radial clearance adjustment. In: Proc. of the 2008 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, New York, USA (2008)
35. Schmidt, U.: Vorspannungsregelung für die Hauptspindellagerung einer Drehmaschine. In: Jahresbericht 2003, Fraunhofer Institut für Werkzeugmaschinen und Umformtechnik, Chemnitz, Germany (2003)
36. Pavlović, N., Keimer, R., Franke, H.-J.: Design of an adaptronic swivel joint for parallel robots based on quasi-statical clearance adjustment. In: Parallel Kinematic Machines in Research and Practice, Proceedings of the 5th Chemnitz Parallel Kinematics Seminar, Chemnitz, Germany, pp. 341–355 (2006)
37. Ersoy, M., Franke, H.-J.: Systematik der Elemente und Operationen bei der konstruktiven Gestaltung. VDI-Z (118/24) (1976)
38. Roth, K.: Konstruieren mit Konstruktionskatalogen, Band 2: Kataloge. Springer, Heidelberg (2001)
39. Pavlović, N., Keimer, R., Franke, H.-J.: Adaptronic revolute joints for parallel robots based on quasi-statical clearance adjustment. In: Proc. of the 17th CISM-IFToMM Symposium ROMANSY - Robot Design, Dynamics and Control, Tokyo, Japan, pp. 459–466 (2008)

Design and Implementation of New Sensors and Their Integration in Joints

Maren Ramona Kirchhoff, Jens Güttler, Alexander Wogersien, Nenad Pavlović, Robert Otremba, Hans-Joachim Franke, and Stephanus Büttgenbach

Abstract. Taking full advantage of the structural capabilities of parallel kinematics demands additional machine-oriented control tasks, e.g. self-calibration and workspace monitoring. A promising strategy to simplify the corresponding algorithms is the integration of lightweight angular position sensors in passive joints. Due to the rough environment, contactless working sensors are required. In this section, two different inductive sensors manufactured by means of microtechnology are presented. The incremental sensor consists of microcoils, whose inductance changes during the rotation of a microscale on the joint shaft. The absolute sensor combines a vectorial magnetometer with a permanent magnet changing its magnetic orientation during joint movement.

1 Introduction

In the field of robotics, position sensor systems are applied for increasing the machine performance and security as well as for including tasks like mechanical system calibration and exception handling routines in the robot control architecture. Laser interferometers or cameras are often installed as external sensors for monitoring or advanced positioning accuracy via image recognition. As internal sensors, linear or angular transducers are mounted, which detect the current position of the robot structure. With the measured data the kinematic models are solved under real-time conditions for implementing the above mentioned machine-oriented control functions. Due to the rough environment within machine components, contactless

Maren Ramona Kirchhoff · Jens Güttler · Alexander Wogersien · Stephanus Büttgenbach
Technische Universität Braunschweig, Institute for Microtechnology,
Alte Salzdahlumer Straße 203, 38124 Braunschweig, Germany
e-mail: {m.kirchhoff,j.guettler,a.wogersien}@tu-bs.de,
s.buettgenbach@tu-bs.de

Nenad Pavlović · Robert Otremba · Hans-Joachim Franke
Technische Universität Braunschweig, Institute for Engineering Design,
Langer Kamp 8, 38106 Braunschweig, Germany
e-mail: {pavlovic,otremba,franke}@ikt.tu-bs.de

D. Schütz and F.M. Wahl (Eds.): Robotic Systems for Handling and Assembly, STAR 67, pp. 445–459.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

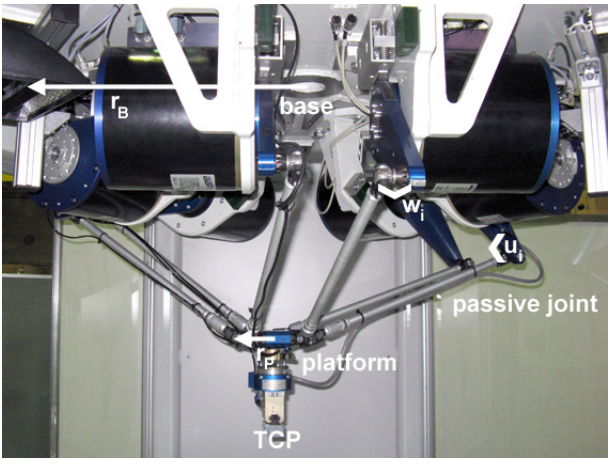


Fig. 1. HEXAII parallel robot.

measuring principles are preferred [1, 2, 3, 4]. State of the art in the field of rotary encoders are optical or Hall Effect sensors. Latest developments in this subject are based on the magneto resistance effect. Such sensors are available highly integrated as surface-mounted devices.

The motivation of the presented work is the full integration of innovative angular position sensors in the passive joints of a parallel kinematic machine (PKM) with HEXA-structure. The HEXAII-robot is one of the demonstrators within the framework of the Collaborative Research Center SFB 562 "Robotic Systems for Handling and Assembly", Braunschweig, Germany [5]. It is investigated for identifying the inherent structural capabilities of PKM and for determining how to take full advantage of these capabilities. Especially vibration suppression within the passive kinematic chains, self-calibration, and tasks for preventing collisions, singularities, and violation of limitations are interesting prospects to assure a save and precise robot movement in the whole workspace. The development of new machine components like adaptive joints and internal angular position sensors and their integration in the HEXAII-robot belong to the ambitious objectives [6, 7, 8].

The HEXAII is a 6 degree of freedom PKM with a base radius of $r_B = 300\,\text{mm}$ and an end effector platform radius of $r_P = 43.3\,\text{mm}$ (see Fig. 1). To solve the algorithms for the direct kinematic model a sensor arrangement referred as 2-2-2 configuration where two sensors are integrated in each of three kinematic chains measuring the angles u_i and w_i is needed [2, 3, 9]. In this way, reliable movements, calculation of the tool center point (TCP) position, and self-calibration of the HEXAII-structure can be performed. Two concepts for angular position sensors based on inductive working principles and their integration in the passive joints of the HEXAII-PKM are discussed in this contribution. The first sensor is an incrementally working one, whereas the second sensor measures absolutely. All sensor components are fabricated by means of micro technologies introduced in the following chapter.

2 Demands on Joint-Sensors

The passive rotary and cardan joints of the HEXAII-PKM are developed at the Institute for Engineering Design, Braunschweig, Germany [10]. They are characterized by low mass and high stiffness to provide high acceleration ($a = 6\,\mathrm{g}$) and high velocity ($v = 5\,\mathrm{m/s}$) of the robot structure. Joint-integrated angular sensors have to be small and lightweight for low weight increase and installation space. Furthermore, high resolution and repeatability are exceedingly important for high performance and accuracy of the parallel robot. A contactless working principle is mandatory for abrasion resistance. In addition all sensor parts should be robust against vibrations and lubricants within the joints. Hence, optical encoders or capacitive transducers are a suboptimal solution for the required application. Innovative concepts for inductively working sensors have been the favored approach within the Collaborative Research Center SFB 562 [6, 7, 8].

The presented sensor systems are manufactured at the Institute for Microtechnology, Braunschweig, Germany. Micro technologies allow the required miniaturization for an easy joint integration. The main micro fabrication batch processes, e.g. lithography, wet chemical or dry etching, doping, and various deposition methods (see Fig. 2), have been established within the field of thin film semiconductor technology.

In contrast, micro systems like lab-on-chip systems, micro motors or grippers, and micro sensors for numerous applications are characterized by three-dimensional structures with high aspect ratios. Silicon, glass, quartz or ceramic wafers build the platform for micro systems. Via lithography the two-dimensional pattern of the required structures are transferred into a photo resist layer deposited onto the

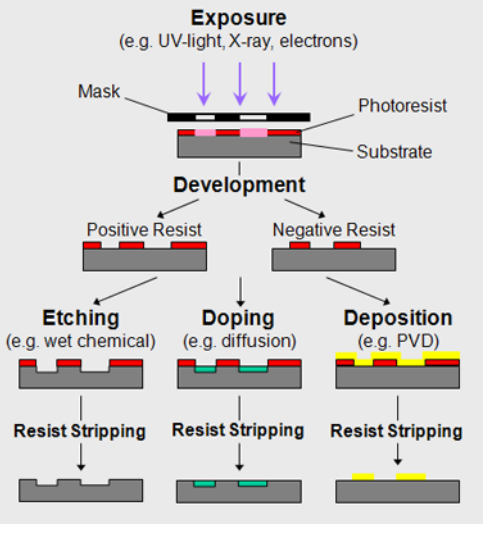


Fig. 2. Basic micro fabrication process sequences.

substrate. Masks for lithographic processes are typically optical quartz glass plates with a structured metal absorber layer. For multilayer devices, mask and already machined substrate are precisely aligned by help of alignment marks. Diffraction effects occuring at ultraviolet radiation (UV) exposure of thick resist layers require an adequate mask design [11, 12]. The exposed resist is developed for dissolving it selectively. Afterwards, the functional layer is manufactured within the resist moulds by substractive or additive processes. Finally the resist is removed using dry or wet stripping processes.

The presented inductively working sensors mainly consist of planar or helical coils with excellent current-carrying copper conductors of several 10 μm cross section. Further constituent parts are insulation layers between the copper conductors and internal magnetic flux guiding cores. UV depth lithography using photo resists with several 10 μm thickness [13, 14] and electro deposition of metal structures are the basic manufacturing processes for these sensors [15, 16, 17]. The electro deposition process of ferromagnetic flux guiding nickel-iron (NiFe) structures has been optimized within the scope of this work [14, 16, 18, 19].

3 Eddy Current Sensor

An angular sensor based on the eddy current feedback effect used for the scanning of metallic increments on a circular micro scale is introduced in this section.

3.1 Functional Principle

Basic part of the eddy current sensor is a planar micro coil located above a disk with circular metallic grating structures. The schematic of this configuration and the planar coil are shown in Fig. 3. The coil is driven by an AC voltage and generates a magnetic field permeating the conductive grating structures vertically and inducing eddy currents in the metallic surfaces. According to Lenz's law, the induced currents generate a contrary magnetic field, which induces a reverse voltage in the coil and changes the coil impedance this way. This so-called eddy current feedback can be detected during the rotation of the micro scale [7, 20]. The amplitude of the sinusoidal measurement signal increases with increasing eddy current feedback and defines the sensitivity of the inductive sensor. The sinusoidal signal is electronically converted to voltage pulses, which are counted. Every pulse corresponds to exactly one grating structure. The counter reading represents the angular range that is passed through during rotation of the micro scale.

Apart from this basic configuration, several more complex designs have been developed with the objective of optimizing the sensor characteristics [2, 3, 21]. Double-layer coils with NiFe cores feature a higher sensor sensitivity due to stronger eddy current feedback. The micro scale has been modified by structuring increments with increased thickness, conductivity, and resolution for identifying correlations between these parameters and the sensor output signal. Furthermore, a second track on the scale allows the detection of a second signal for recognizing direction

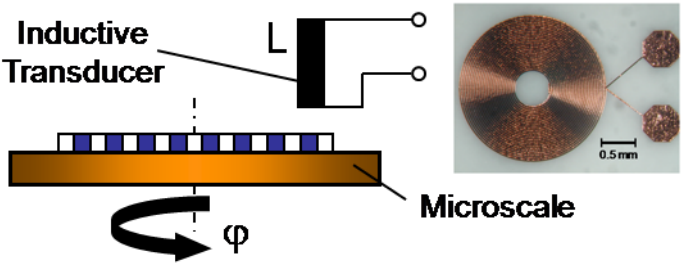


Fig. 3. Sensor configuration with planar copper micro coil.

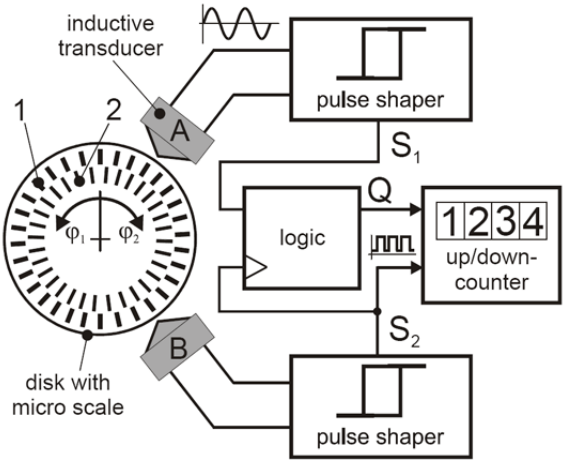


Fig. 4. Schematic of signal processing with two tracks.

reversals. This configuration is shown in Fig. 4. The phase shift between the signals S_1 and S_2 is analyzed and the direction of counting the pulses generated by track 1 changes immediately, when signal S_2 changes from lagging to leading signal S_1.

3.2 Experimental Results

The test-setup for characterizing the incremental sensor is shown in Fig. 5. The testing scale features several tracks and is fixed on a rotary stage driven by a DC motor with a resolution of 0.001°. The sensor is fixed on a linear stage for controlling the distance between coil and testing scale. The eddy current feedback increases with decreasing distance.

Fig. 6 shows the sensor characteristic at 10 MHz excitation frequency. Three tracks have been scanned, whereas track 1 features the lowest resolution (4.5°) and causes the highest output amplitude. An increasing resolution of the micro scale results in decreasing sensor sensitivity [20, 21].

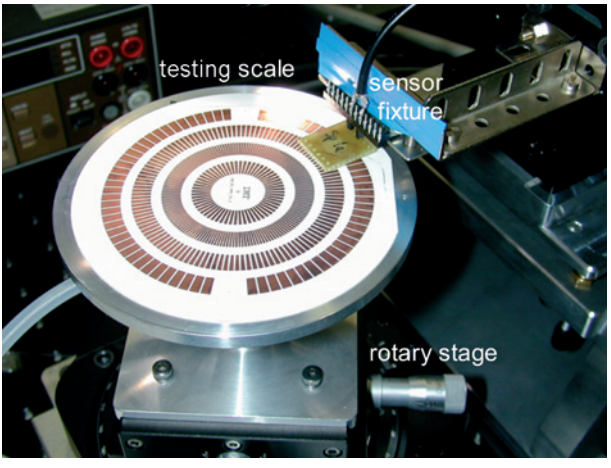


Fig. 5. Test-setup with testing scale on a precise rotary stage.

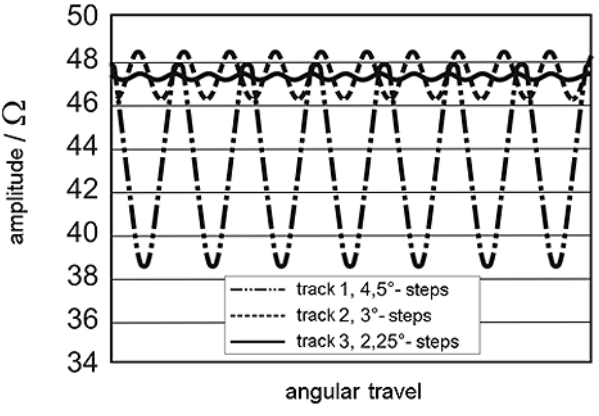


Fig. 6. Sensor output signal for three scale resolutions.

3.3 Integration

The integration of sensors adds challenging boundary conditions to the passive HEXAII-joint development [22, 23, 24]. A housing for protecting the sensor from environmental influences as well as from inherent impacts like lubricants and wear debris is needed. Accelerated tests of the eddy current sensor have been performed without full integration. The semi-integration of a sensor with single coil configuration, a micro scale resolution of 2°, and a measuring range of 140° in passive cardan HEXAII-joints is shown in Fig. 7. This prototype has been successfully tested at a robot velocity of $v = 1.2\,\mathrm{m/s}$ [3, 23, 24].

Due to the rough environment, a full sensor integration with closed sensor housings protecting the planar coil and the micro scale is preferred. These housings have

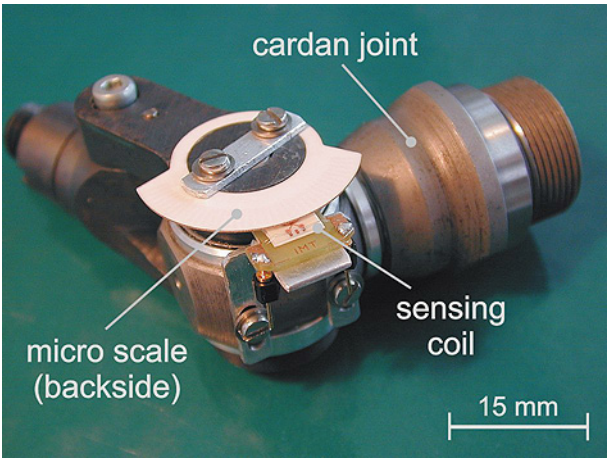


Fig. 7. Semi-integration of eddy current sensor in passive HEXAII-joint.

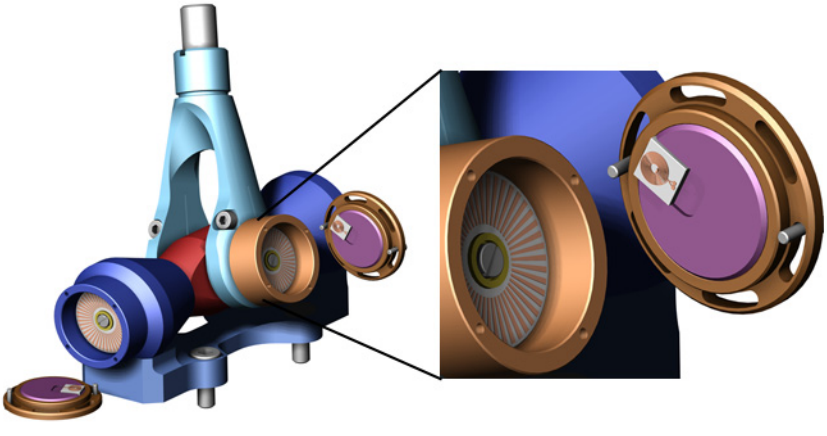


Fig. 8. Schematic: full sensor integration in passive HEXAII-joint.

to be small-sized to minimize the risk of collisions between any structural parts of the kinematic chains. Further geometrical constraints have to be taken into account. The presented eddy current sensor is composed of two flat parallel parts, namely planar coil and micro scale. The rotation around an axis perpendicular to these flat parts is analyzed for measuring the joint angle, and the sensor has to be aligned with respect to this joint axis. A joint comprising such sensors has to be build as a serial combination of two rotary joints whose axes are not parallel to each other and intersect in one point. A joint with two axes acts as an universal joint. The addition of a further axis leads to a ball joint.

In order to avoid a compromise concerning the stiffness and load capacity, the sensor itself should be placed outside the load carrying area of the joint. This leads

to a joint arrangement were the hollow parts of the axes (i.e. the bearing housings) are placed on the outer sides and the shafts are placed inside. The schematic view in Fig. 8 shows a possible configuration, where the micro scales are fixed onto the two joint shafts rotating during robot movement.

The planar coils are fixed at the housing cap. This complex redesign of the HEX-AII-joints has not been realized for eddy current sensor integration due to the low sensor resolution, that could be achieved (cp. 3.2). The next section discusses an absolutely measuring sensor with higher resolution. New joints supporting the concept described above are presented in 4.3.

4 Fluxgate Magnetometer

An absolutely measuring sensor based on the inductively working fluxgate principle has been miniaturized and adapted for full joint-integration by means of micro technologies. Details about this sensor are discussed in the following section.

4.1 Functional Principle

Fluxgates are vectorial magnetic field sensors commonly used for geodetic or aerospace-related measurements. Comprehensive information about fluxgates can be found in [25, 26, 27, 28]. Their directional sensitivity regarding a DC magnetic field vector is adopted for realizing an angular joint-sensor in the scope of this work. Therefore, the magnetometer is combined with a miniaturized permanent magnet.

The configuration of a fluxgate based on helical coils around a ferromagnetic core is shown in Fig. 9. The NiFe core is periodically saturated by 20 kHz voltage excitation. The excited magnetic flux induces voltage peaks in the sensing coil. The induced signal is modified by superposition of internal and external magnetic field vectors at core level. The 2^{nd} harmonic is the signal part of the sensed AC voltage, which corresponds with an external magnetic field vector according to Fourier transform theory and carries the required information about the field direction.

Apart from the ring-core fluxgate presented in Fig. 9 and described in [6, 16, 29], a second core geometry called racetrack has been realized (see Fig. 10). The parallel core sections feature a better directional sensitivity. The induced voltage signal in the sensing coil is zero, when the external field vector is vertical to the parallel core sections. If the field direction is parallel to these sections, the signal reaches maximum values [8, 19, 30, 31].

4.2 Experimental Results

Generation of the excitation voltage and analysis of the 2^{nd} harmonic are performed by the electronic circuit shown in Fig. 11 [19, 32]. The main device is the microcontroller for generating the excitation signal and a reference signal with double excitation frequency. The signals have to be free from phase shift for an excellent

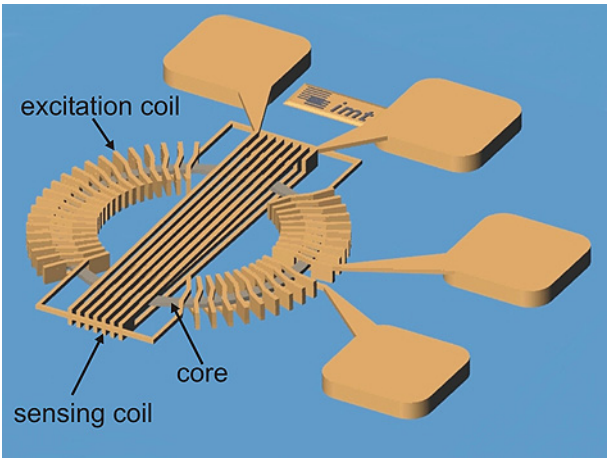


Fig. 9. Schematic: configuration of a micro fluxgate magnetometer.

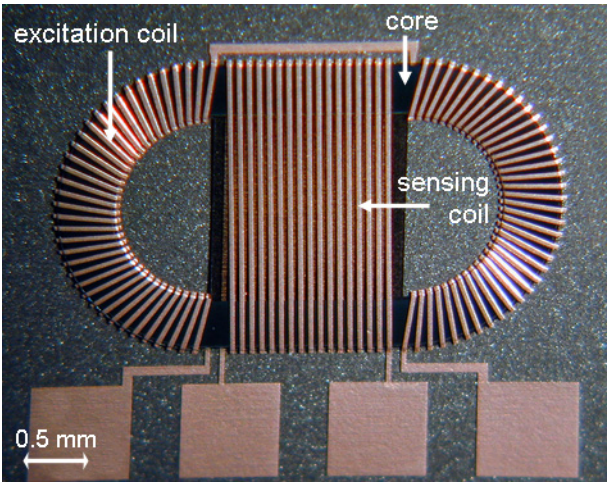


Fig. 10. Micro fluxgate with racetrack-core.

phase selective rectifying of the bandpass filtered 2^{nd} harmonic. The amplified and filtered sensor signal is connected to a 16 bit A/D converter. The HEXAII-robot controller unit, which employs a power supply of ±12 V, also powers the presented fluxgate electronic.

The test-setup for characterizing the fluxgate sensor (see Fig. 12) is very similiar to the setup for the eddy current sensor described in 3.2. The sensor fixture can be adjusted with a resolution of 1 µm in xyz directions above the permanent magnet, which is located on a rotary platform.

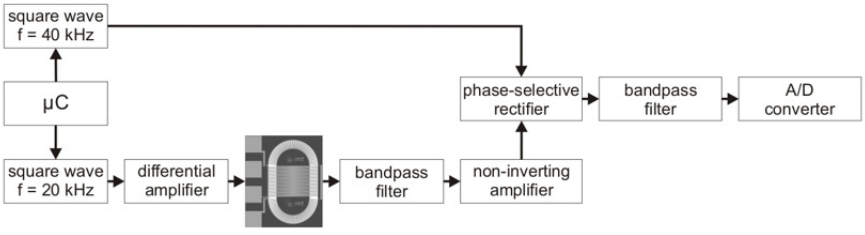


Fig. 11. Electrical circuit for driving and analyzing the fluxgate.

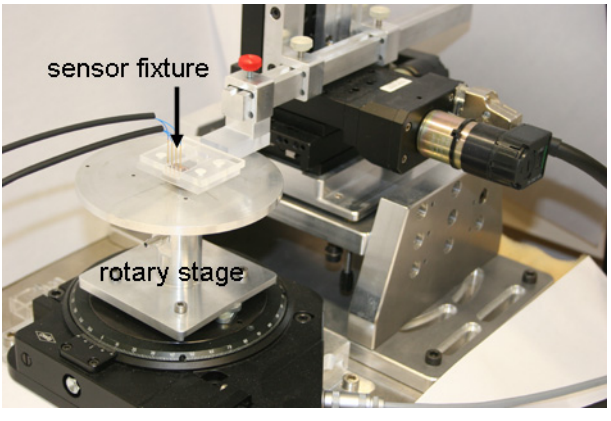


Fig. 12. Test-setup with permanent magnet on a precise rotary stage.

Fig. 13 shows the characteristic of the racetrack fluxgate. Maximum output voltage values are +5 V and −5 V, respectively. These values are reached at 90° and 270°, which mean a parallel orientation of the field vector relative to the parallel core sections. Zero-crossing of the output voltage occures at 180°. In the linear span of the characteristic (between 150° and 210°) a sensor sensitivity of 113 mV/° is reached. After 16 bit A/D conversion of this 60° measurement range, a resolution of 0.001° is theoretically performed. Parasitic induction and noise effects caused by surrounding electro-magnetic fields impair this theoretical value. The calculated resolution based on the measured data amounts < 0.2°.

4.3 Integration

The full joint-integration of the fluxgate magnetometer has been realized in cooperation between the Institute for Engineering Design and the Institute for Microtechnology, Braunschweig, Germany [6, 8, 31]. The concept described in 3.3 has been adapted to the development of novel joints with a suitable sensor housing protecting the fluxgate and the permanent magnet. As can be seen at the schematic view in Fig. 14, the permanent magnets are fixed onto the joint shafts and rotate during joint

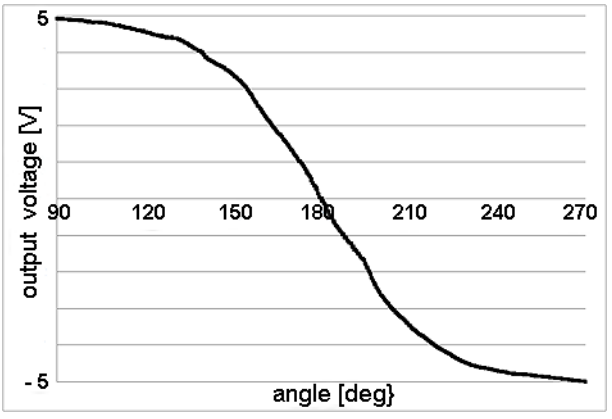


Fig. 13. Characteristic of the racetrack fluxgate.

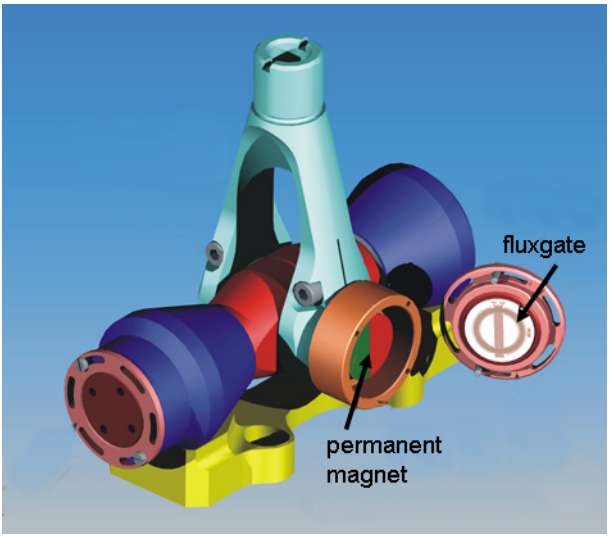


Fig. 14. Schematic: full fluxgate integration.

movement similar to the micro scales in Fig. 8. The sensors are located inside the housing caps.

The manufactured cardan HEXAII-joints are presented in Fig. 15 [6, 8, 23, 24, 31]. They can be fabricated in standard versions without sensor housing (Fig. 15, left side) as well as in modified versions allowing the integration of sensors (Fig. 15, right side). The modification consists of replacing the shafts and bearing modules and adding additional caps. The joint is dimensioned to withstand a load up

Fig. 15. HEXAII-joint: standard version (left) and modified version (right).

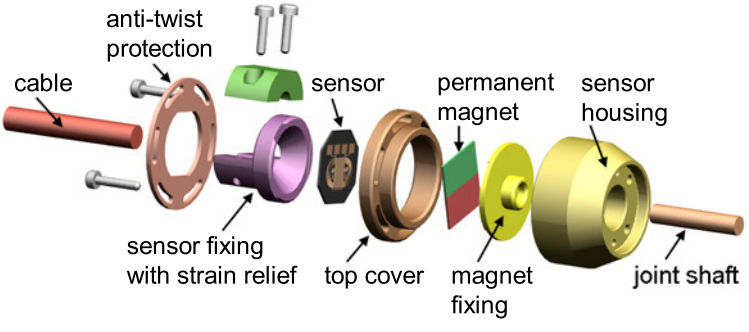


Fig. 16. Exploded view of the sensor housing.

to 2.5 kN. The pivoting angles are 360° and ±133°, respectively. The joint weighs 203 g in the standard version. The position of the bearing modules can be adjusted to compensate tolerances. The stiffness in the movement range used with the HEXAII (47° - 133°) ranges from 8.3 N/µm to 12 N/µm.

Fig. 16 shows an exploded view of the sensor housing (cp. Fig. 15, right side). The permanent magnet is located in the magnet fixing which rotates on the joint shaft. The fluxgate magnetometer is mounted into the milled cavity (13 x 14 mm) of the sensor fixture. The fixture features a strain relief for the four-wire sensor cable and an outside fine pitch thread. The latter is screwed into the top cover for calibrating the distance between sensor and magnet. The maximum distance amounts 1.5 mm. For calibrating the sensor orientation relative to the magnet, the top cover is equipped with slotted holes for fixing it onto the sensor housing. An anti-twist protection ring provides stiffness during robot movement.

5 Results

Two inductively working angular sensor concepts have been developed and examined with respect to parallel robot application. The functional principles of both sensors are suited for angular measurements in the range of 180°. All sensor parts have been adopted for joint integration by means of micro technologies, whereas several fabrication processes have been implemented and optimized for reliable and efficient manufacturing. Adequate joint modifications including optimized design, stiffness and protective sensor housings have been put in operation at the HEXAII-robot successfully.

The electrical circuit for generating the fluxgate excitation signal and processing the sensor output is much more sophisticated in comparison to the circuit for driving the incremental sensor. On the other hand, the absolutely measuring fluxgate magnetometer achieves the best results regarding resolution and sensitivity in comparison to the incremental eddy current sensor. Especially the racetrack core shape significantly increases the sensitivity and provides a resolution $< 0.2°$.

The required accuracy at the TCP of the 6 degree of freedom HEXAII-PKM has been chosen in [2, 3, 9] as $\triangle \boldsymbol{TCP}$ = [0.5mm 0.5mm 0.5mm 0.5° 0.5° 0.5°]. By taking these values and the 2-2-2 angular joint-sensor configuration as a basis, a necessary sensor resolution of 0.05° has been calculated. The theoretical racetrack fluxgate resolution of 0.003° undercuts the required value. Further developments regarding an effective shielding is needed to reach the required performance. However, commercially available angular sensors featuring the required resolution are inapplicable for full joint integration.

6 Conclusion

Inductively working sensor systems for angular position measurements are presented in this work. The sensors are included in the passive cardan joints of a HEXAII parallel robot, which has been developed within the framework of the Collaborative Research Center SFB 562, Braunschweig, Germany. The measured data simplifies the solution of direct kinematics and features the implementation of machine-oriented control tasks like workspace monitoring and self-calibration. For a full joint integration, the sensors have been manufactured using micro technology processes based on UV lithography and electro deposition. The cardan joints have been modified by designing a protective sensor housing with calibration mechanism.

Acknowledgements. This work is funded by the German Research Foundation within the scope of the Collaborative Research Center SFB 562 "Robotic Systems for Handling and Assembly".

References

1. Boese, C., Kirchhoff, M.R., Feldmann, M., Güttler, J., Büttgenbach, S.: New generation of integrated position sensor systems for parallel robotic applications. In: IEEE Sensors 2009, pp. 1362–1365 (2009) ISBN 978-1-4244-5335-1

2. Hesselbach, J., Bier, C., Pietsch, I., Plitea, N., Büttgenbach, S., Wogersien, A., Güttler, J.: Passive joint-sensor applications for parallel robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 4, pp. 3507–3512 (2004)
3. Hesselbach, J., Bier, C., Pietsch, I., Plitea, N., Büttgenbach, S., Wogersien, A., Güttler, J.: Passive-joint sensors for parallel robots. Mechatronics 15(1), 43–65 (2005)
4. Kirchhoff, M.R., Boese, C., Güttler, J., Feldmann, M., Büttgenbach, S.: Innovative high-precision position sensor systems for robotic and automotive applications. Procedia Chemistry 1(1), 501–504 (2009)
5. Schütz, D., Budde, C., Raatz, A., Hesselbach, J.: Parallel Kinematic Structures of the SFB 562. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 109–124. Springer, Heidelberg (2010)
6. Büttgenbach, S., Güttler, J., Last, P., Bier, C., Otremba, R.: Development of angular joint-sensors and application to parallel robots. In: Last, P. (ed.) 2nd Int. Colloquium of the Collaborative Research Center SFB 562, Shaker, Aachen. Fortschritte in der Robotik, vol. 9, pp. 237–251 (2005) ISBN 3-8322-3866-2
7. Büttgenbach, S., Wogersien, A., Bier, C., Pietsch, I.T.: Microsensors for parallel robots. In: Krefft, M. (ed.) First Int. Colloquium of the Collaborative Research Center SFB 562. Fortschritte in der Robotik, vol. 7, pp. 141–152. Shaker, Aachen (2002)
8. Franke, H.-J., Pavlović, N., Kirchhoff, M.R., Büttgenbach, S.: Integration of sensors and actuators in joints of parallel robots. In: Schütz, D., Raatz, A., Wahl, F.M. (eds.) 3rd Int. Colloquium of the Collaborative Research Center SFB 562. Fortschritte in der Robotik, vol. 14, pp. 169–180. Shaker, Aachen (2008)
9. Last, P., Budde, C., Bier, C., Hesselbach, J.: Hexa-parallel-structure calibration by means of angular passive joint sensors. In: IEEE Int. Conf. on Mechatronics and Automation, vol. 3, pp. 1300–1305 (2005)
10. Pavlović, N., Otremba, R., Inkermann, D., Franke, H.J., Vietor, T.: Passive and adaptive joints for parallel robots. In: Schütz, D., Wahl, F.M. (eds.) Robotic Systems for Handling and Assembly. STAR, vol. 67, pp. 429–444. Springer, Heidelberg (2010)
11. Triltsch, U., Feldmann, M., Büttgenbach, S.: Optimization of masks for high aspect ratio UV-lithographic patterning. In: Laudon, M. (ed.) NSTI Nanotechnology Conference and Trade Show, pp. 81–84. NSTI [u.a.], Cambridge (2007) ISBN 1420061844
12. Triltsch, U., Feldmann, M., Boese, C., Büttgenbach, S.: Simulation tool for proximity effects in high aspect ratio UV-lithographic patterning. Sensors and Actuators, A: Physical 142(1), 429–433 (2008)
13. Büttgenbach, S., Feldmann, M.: Application of UV depth lithography in micro system technology. Optoelectronics Letters 4(1), 1–4 (2008)
14. Feldmann, M.: Technologien und Applikationen der UV-Tiefenlithographie: Mikroaktorik, Mikrosensorik und Mikrofluidik. Berichte aus der Mikro-und Feinwerktechnik, vol. 20. Shaker, Aachen (2007) ISBN 978-3-8322-6146-7
15. Feldmann, M., Waldschik, A., Büttgenbach, S.: Technology and application of electrodepositable photo resists to create uniform coatings needed for complex 3d micro actuators and sensors. Microsystem Technologies 13(5-6), 557–562 (2007)
16. Güttler, J.: Entwicklung eines gelenkintegrierten Winkelsensors für den Einsatz in hochdynamischen Parallelrobotern. Berichte aus der Mikro-und Feinwerktechnik, vol. 19. Shaker, Aachen (2006) ISBN 3-8322-5681-4
17. Kirchhoff, M.R., Güttler, J., Waldschik, A., Feldmann, M., Büttgenbach, S.: Revised fabrication process for micro-fluxgate-magnetometers: Usage of electrodepositable photoresist. Microelectronic Engineering 85(5-6), 1047–1049 (2008)

18. Jordan, A., Kirchhoff, M.R., Büttgenbach, S.: Effect analysis of magnetic annealing below curie-temperature on the magnetic properties of electrodeposited nickel-iron. Microelectronic Engineering 87, 1223–1225
19. Kirchhoff, M.R., Bogdanski, G., Büttgenbach, S.: Design, fabrication and characterization of a micro-fluxgate intended for parallel robot application. In: Proc. of SPIE - The International Society for Optical Engineering, vol. 7362 (2009)
20. Wogersien, A., Beißner, S., Samson, S., Büttgenbach, S.: Development of new inductive microsensors for angular position measurement. In: Proc. of the 2nd VDE World Microtechnologies Congress, pp. 159–163 (2003)
21. Wogersien, A., Samson, S., Güttler, J., Beißner, S., Büttgenbach, S.: Novel inductive eddy current sensor for angle measurement. In: IEEE Sensors 2003, vol. 2(1), pp. 236–241 (2003)
22. Franke, H.-J., Otremba, R., Jänicke, T.: Methodical development of optimized passive joints. In: Krefft, M., Wahl, F.M. (eds.) First Int. Colloquium of the Collaborative Research Center SFB 562. Fortschritte in der Robotik, vol. 7, pp. 119–130. Shaker, Aachen (2002)
23. Otremba, R.: Methodische Entwicklung von Gelenken für Parallelroboter. Konstruktion (10), 82–86 (2005)
24. Otremba, R.: Systematische Entwicklung von Gelenken für Parallelroboter. Bericht / Inst. f. Konstruktionstechnik, vol. 67. Techn. Univ. Braunschweig, Logos, Berlin (2005) ISBN 3-8325-0811-2
25. Primdahl, F.: The fluxgate magnetometer. Journal of Physics E: Scientific Instruments 12(4), 241–253 (1979)
26. Primdahl, F., Hernando, B., Nielsen, O.V., Petersen, J.R.: Demagnetising factor and noise in the fluxgate ring-core sensor. Journal of Physics E: Scientific Instruments 22(12), 1004–1008 (1989)
27. Ripka, P.: Review of fluxgate sensors. Sensors and Actuators: A. Physical 33(3), 129–141 (1992)
28. Ripka, P.: Advances in fluxgate sensors. Sensors and Actuators, A: Physical 106(1-3), 8–14 (2003)
29. Güttler, J., Last, P., Otremba, R., Büttgenbach, S.: A novel angular joint-sensor using a fluxgate magnetometer. In: IEEE Sensors 2005, pp. 53–56 (2005)
30. Kirchhoff, M.R., Güttler, J., Büttgenbach, S.: Mikro-Magnetometer nach dem Fluxgate-Prinzip für Anwendungen in der Roboter-Sensorik: Design und mikrotechnologischer Fertigungsprozess. In: Geßner, T. (ed.) Proc. Mikrosystemtechnik Kongress, pp. 959–962. VDE, Berlin (2007) ISBN 978-3-8007-3061-2
31. Kirchhoff, M.R., Pavlović, N., Franke, H.-J., Büttgenbach, S.: Integration eines mikrotechnisch hergestellten Magnetometers in passive Gelenke von Parallelrobotern: Erhöhung der Funktionalität von Maschinenkomponenten. VDI Berichte (2012), 313–316 (2008)
32. Kirchhoff, M.R., Büttgenbach, S.: MEMS fluxgate magnetometer for parallel robot application. Journal = Microsystem Technologies 16(5), 787–790 (2010)

Author Index

Algermissen, Stephan 159

Beckmann, Guido 213
Bier, Carlos 77
Budde, Christoph 109, 175
Büttgenbach, Stephanus 445

Dadji, Yannick 213
Diethers, Karsten 373
Dietrich, Franz 77, 315

Finkemeyer, Bernd 193, 293
Franke, Hans-Joachim 59, 429, 445
Frindt, Matthias 17

Goltz, Ursula 355, 373
Güttler, Jens 445

Hagner, Matthias 373
Helm, Manfred 175
Hesselbach, Jürgen 17, 77, 93, 109, 175, 315

Inkermann, David 429

Keimer, Ralf 413
Kirchhoff, Maren Ramona 445
Kock, Sönke 143
Kohn, Nnamdi 213
Kolbus, Michael 233, 253
Krefft, Mathias 17
Kröger, Torsten 193, 275, 293

Last, Philipp 93, 175

Maaß, Jochen 77, 315, 355
Michalik, Harald 213
Möglich, Tobias 213

Osypiuk, Rafal 275
Otremba, Robert 429, 445

Pavlović, Nenad 429, 445
Pietsch, Ingo 77

Raatz, Annika 77, 93, 109, 175, 315
Reisinger, Thomas 233, 253
Rose, Michael 39, 393

Schreiber, Frank 125
Schumacher, Walter 125, 143, 233, 253
Schütz, Daniel 109
Sinapius, Michael 159, 413
Stachera, Krzysztof 125
Stechert, Carsten 59
Steiner, Jens 213, 355, 373

Thomas, Ulrike 3, 333

Varchmin, Jörn-Uwe 213
Vietor, Thomas 59, 429

Wahl, Friedrich M. 3, 193, 293, 333
Wobbe, Frank 233, 253
Wogersien, Alexander 445

Springer Tracts in Advanced Robotics

Edited by B. Siciliano, O. Khatib and F. Groen

Further volumes of this series can be found on our homepage: springer.com

Vol. 66: Kaneko, M.; Nakamura, Y. (Eds.)
Robotics Research
450 p. 2010 [978-3-642-14742-5]

Vol. 65: Ribas, D.; Ridao, P.; Neira, J.
Underwater SLAM for Structured
Environments Using an
Imaging Sonar
142 p. 2010 [978-3-642-14039-6]

Vol. 64: Vasquez Govea, A.D.
Incremental Learning for Motion Prediction
of Pedestrians and Vehicles
153 p. 2010 [978-3-642-13641-2]

Vol. 63: Vanderborght, B.;
Dynamic Stabilisation of the
Biped Lucy Powered by Actuators
with Controllable Stiffness
281 p. 2010 [978-3-642-13416-6]

Vol. 62: Howard, A.; Iagnemma, K.;
Kelly, A. (Eds.):
Field and Service Robotics
511 p. 2010 [978-3-642-13407-4]

Vol. 61: Mozos, Ó.M.
Semantic Labeling of Places with
Mobile Robots
134 p. 2010 [978-3-642-11209-6]

Vol. 60: Zhu, W.-H.
Virtual Decomposition Control –
Toward Hyper Degrees of
Freedom Robots
443 p. 2010 [978-3-642-10723-8]

Vol. 59: Otake, M.
Electroactive Polymer Gel Robots –
Modelling and Control of Artificial Muscles
238 p. 2010 [978-3-540-23955-0]

Vol. 58: Kröger, T.
On-Line Trajectory Generation in Robotic
Systems – Basic Concepts for Instantaneous
Reactions to Unforeseen (Sensor) Events
230 p. 2010 [978-3-642-05174-6]

Vol. 57: Chirikjian, G.S.; Choset, H.;
Morales, M., Murphey, T. (Eds.)
Algorithmic Foundations
of Robotics VIII – Selected Contributions
of the Eighth International Workshop on the
Algorithmic Foundations of Robotics
680 p. 2010 [978-3-642-00311-0]

Vol. 56: Buehler, M.; Iagnemma, K.;
Singh S. (Eds.)
The DARPA Urban Challenge – Autonomous
Vehicles in City Traffic
625 p. 2009 [978-3-642-03990-4]

Vol. 55: Stachniss, C.
Robotic Mapping and Exploration
196 p. 2009 [978-3-642-01096-5]

Vol. 54: Khatib, O.; Kumar, V.;
Pappas, G.J. (Eds.)
Experimental Robotics:
The Eleventh International Symposium
579 p. 2009 [978-3-642-00195-6]

Vol. 53: Duindam, V.; Stramigioli, S.
Modeling and Control for Efficient Bipedal
Walking Robots
211 p. 2009 [978-3-540-89917-4]

Vol. 52: Nüchter, A.
3D Robotic Mapping
201 p. 2009 [978-3-540-89883-2]

Vol. 51: Song, D.
Sharing a Vision
186 p. 2009 [978-3-540-88064-6]

Vol. 50: Alterovitz, R.; Goldberg, K.
Motion Planning in Medicine: Optimization
and Simulation Algorithms for
Image-Guided Procedures
153 p. 2008 [978-3-540-69257-7]

Vol. 49: Ott, C.
Cartesian Impedance Control of Redundant
and Flexible-Joint Robots
190 p. 2008 [978-3-540-69253-9]

Vol. 48: Wolter, D.
Spatial Representation and
Reasoning for Robot
Mapping
185 p. 2008 [978-3-540-69011-5]

Vol. 47: Akella, S.; Amato, N.;
Huang, W.; Mishra, B.; (Eds.)
Algorithmic Foundation of Robotics VII
524 p. 2008 [978-3-540-68404-6]

Vol. 46: Bessière, P.; Laugier, C.;
Siegwart R. (Eds.)
Probabilistic Reasoning and Decision
Making in Sensory-Motor Systems
375 p. 2008 [978-3-540-79006-8]

Vol. 45: Bicchi, A.; Buss, M.;
Ernst, M.O.; Peer A. (Eds.)
The Sense of Touch and Its Rendering
281 p. 2008 [978-3-540-79034-1]

Vol. 44: Bruyninckx, H.; Přeučil, L.;
Kulich, M. (Eds.)
European Robotics Symposium 2008
356 p. 2008 [978-3-540-78315-2]

Vol. 43: Lamon, P.
3D-Position Tracking and Control
for All-Terrain Robots
105 p. 2008 [978-3-540-78286-5]

Vol. 42: Laugier, C.; Siegwart, R. (Eds.)
Field and Service Robotics
597 p. 2008 [978-3-540-75403-9]

Vol. 41: Milford, M.J.
Robot Navigation from Nature
194 p. 2008 [978-3-540-77519-5]

Vol. 40: Birglen, L.; Laliberté, T.; Gosselin, C.
Underactuated Robotic Hands
241 p. 2008 [978-3-540-77458-7]

Vol. 39: Khatib, O.; Kumar, V.; Rus, D. (Eds.)
Experimental Robotics
563 p. 2008 [978-3-540-77456-3]

Vol. 38: Jefferies, M.E.; Yeap, W.-K. (Eds.)
Robotics and Cognitive Approaches to
Spatial Mapping
328 p. 2008 [978-3-540-75386-5]

Vol. 37: Ollero, A.; Maza, I. (Eds.)
Multiple Heterogeneous Unmanned Aerial
Vehicles
233 p. 2007 [978-3-540-73957-9]

Vol. 36: Buehler, M.; Iagnemma, K.;
Singh, S. (Eds.)
The 2005 DARPA Grand Challenge – The Great
Robot Race
520 p. 2007 [978-3-540-73428-4]

Vol. 35: Laugier, C.; Chatila, R. (Eds.)
Autonomous Navigation in Dynamic
Environments
169 p. 2007 [978-3-540-73421-5]

Vol. 33: Kong, X.; Gosselin, C.
Type Synthesis of Parallel
Mechanisms
272 p. 2007 [978-3-540-71989-2]

Vol. 34: Wisse, M.; van der Linde, R.Q.
Delft Pneumatic Bipeds
136 p. 2007 [978-3-540-72807-8]

Vol. 30: Brugali, D. (Ed.)
Software Engineering for Experimental Robotics
490 p. 2007 [978-3-540-68949-2]

Vol. 29: Secchi, C.; Stramigioli, S.; Fantuzzi, C.
Control of Interactive Robotic Interfaces – A
Port-Hamiltonian Approach
225 p. 2007 [978-3-540-49712-7]

Vol. 28: Thrun, S.; Brooks, R.;
Durrant-Whyte, H. (Eds.)
Robotics Research – Results of the 12th
International Symposium ISRR
602 p. 2007 [978-3-540-48110-2]

Vol. 27: Montemerlo, M.; Thrun, S.
FastSLAM – A Scalable Method for the
Simultaneous Localization and Mapping
Problem in Robotics
120 p. 2007 [978-3-540-46399-3]

Vol. 26: Taylor, G.; Kleeman, L.
Visual Perception and Robotic Manipulation – 3D
Object Recognition, Tracking and Hand-Eye
Coordination
218 p. 2007 [978-3-540-33454-5]

Vol. 25: Corke, P.; Sukkarieh, S. (Eds.)
Field and Service Robotics – Results of the 5th
International Conference
580 p. 2006 [978-3-540-33452-1]

Vol. 24: Yuta, S.; Asama, H.; Thrun, S.;
Prassler, E.; Tsubouchi, T. (Eds.)
Field and Service Robotics – Recent Advances in
Research and Applications
550 p. 2006 [978-3-540-32801-8]